



**Titre:** Modélisation et simulation thermique des circuits intégrés  
Title: tridimensionnels

**Auteur:** Amira Aouina  
Author:

**Date:** 2017

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Aouina, A. (2017). Modélisation et simulation thermique des circuits intégrés  
Citation: tridimensionnels [Mémoire de maîtrise, École Polytechnique de Montréal].  
PolyPublie. <https://publications.polymtl.ca/2711/>

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/2711/>  
PolyPublie URL:

**Directeurs de  
recherche:** Gabriela Nicolescu, Giovanni Beltrame, & Luc Fréchette  
Advisors:

**Programme:** Génie informatique  
Program:

UNIVERSITÉ DE MONTRÉAL

MODÉLISATION ET SIMULATION THERMIQUE DES CIRCUITS INTÉGRÉS  
TRIDIMENSIONNELS

AMIRA AOUINA

DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE INFORMATIQUE)

AOÛT 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

MODÉLISATION ET SIMULATION THERMIQUE DES CIRCUITS INTÉGRÉS  
TRIDIMENSIONNELS

présenté par : AOUINA Amira

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. BOIS Guy, Ph. D., président

Mme NICOLESCU Gabriela, Doctorat, membre et directrice de recherche

M. BELTRAME Giovanni, Ph. D., membre et codirecteur de recherche

M. FRÉCHETTE Luc, Ph. D., membre et codirecteur de recherche

M. GUIBAULT François, Ph. D., membre

## **DÉDICACE**

Je dédie ce travail à mes enfants, mes parents, ma famille et mes amis et à toute personne qui m'a soutenue et a cru en moi.

## REMERCIEMENTS

Je tiens à remercier ma directrice de recherche professeur Gabriela Nicolescu et mes codirecteurs professeurs Giovanni Beltrame et Luc Fréchette pour leur encadrement et leur aide.

Je remercie mes collègues pour leur support, encouragement, aide et toutes les belles choses partagées dans notre laboratoire.

Je remercie mes professeurs du département génie informatique et génie logiciel pour toutes les connaissances que j'ai acquises au cours de cette maîtrise.

Je remercie le personnel du département génie informatique et génie logiciel pour son aide et sa patience.

Je remercie ma famille et mes amis pour leur soutien et leur support.

Enfin, je remercie la vie pour tous les défis qu'elle nous lance, les secondes chances qu'elle nous offre et les opportunités qu'elle nous donne.

## RÉSUMÉ

Depuis leur apparition, les circuits intégrés connaissent une évolution exponentielle dans de nombreux axes. Les circuits intégrés tridimensionnels (3D) gagnent de l'intérêt sur le marché en raison d'une multitude d'avantages comme une taille réduite, un grand nombre de fonctionnalités, une faible consommation d'énergie et une bande passante élevée. Mais en raison de leur architecture tridimensionnelle basée sur une superposition de couches multiples dans une dimension compacte, ces circuits présentent de nouveaux défis en termes de contrôle de température. Le recours à la simulation thermique est incontournable pour permettre de prédire la température et aider le concepteur de ce type de circuit à évaluer de multiples alternatives avant la phase de fabrication. Une simulation efficace doit être précise, facile à réaliser, générique et doit prendre en compte l'aspect évolutif des circuits. Les simulateurs thermiques actuels sont coûteux, gourmands en mémoire et peinent à modéliser les futurs circuits intégrés et à suivre leur évolution.

Ce mémoire introduit ICTherm : un outil de simulation thermique précis, générique, code ouvert et facile à utiliser. ICTherm est basé sur un ensemble de solveurs numériques et de méthodes d'optimisation assurant une simulation efficace pour les états permanents et transitoires. Une implémentation flexible de l'approche proposée assure la portabilité du code et la facilité de la mise à jour de l'outil. Une interface utilisateur ergonomique est proposée pour permettre une représentation simple des données à saisir ainsi qu'un module de sortie graphique facilitant l'interprétation des résultats pour les utilisateurs. Un ensemble de modules et d'options est proposé pour assurer la modélisation de tout type de circuit intégré tridimensionnel.

Nous avons validé l'outil ICTherm proposé à travers divers problèmes thermiques représentatifs. La précision des résultats est semblable à celle des outils commerciaux comme Comsol et nos résultats montrent la capacité de l'outil à modéliser et à simuler de futurs circuits avec de nouvelles contraintes comme les circuits optiques.

## ABSTRACT

Since their appearance, the integrated circuit experiencing an exponential evolution in many axis. The three-dimensional integrated circuits are a class of those new ICs that gain more and more interest in many market fields due to their considerable benefits as small size, big number of functionalities, low power consumption and high bandwidth. Because of this 3D architecture based on a superposition of multiple layers in a compact dimension, the 3D ICs carries new challenges in terms of design complexity, devices testing and effective temperature control. Making an efficient design of such devices require multi-discipline skills in electrical, mechanical and thermodynamic fields which constitute a big challenge for the designers. Therefore new, efficient thermal modelling and simulation solutions are required.

Thermal modelling and simulation of an integrated circuit aims to reproduce its thermal behaviour in order to predict its components temperature. An efficient heat transfer simulation of the circuit allows to predict the heat and help the designer to evaluate multiple alternatives while the layout and power output of the device are still subject to modifications.

This work present ICTherm: an efficient, accurate, generic, open source and easy to use thermal simulation tool. ICTherm is based on a set of numerical solvers and optimization methods ensuring an efficient simulation for both steady and transient states. Flexible software implementation of the proposed approach is offered ensuring the code portability and the ease upgrade of the tool. An ergonomic user interface is proposed as a representation of the needed inputs; like circuit architecture, geometries, simulation parameters; and the thermal schema output to facilitate the use of the simulation tool and results interpretation for users. A set of modules and options are proposed to increase the ability to model any type of three-dimensional integrated circuits.

We evaluated the tool on a various representative thermal problems. The accuracy of the results is similar to commercial state of art tools like Comsol and our results show up the capacity of the tool to model and simulate future circuits with new constraints like optical circuits.

## TABLE DES MATIÈRES

DÉDICACE.....	III
REMERCIEMENTS .....	IV
RÉSUMÉ.....	V
ABSTRACT .....	VI
TABLE DES MATIÈRES .....	VII
LISTE DES TABLEAUX.....	XI
LISTE DES FIGURES.....	XII
LISTE DES ALGORITHMES.....	XIV
LISTE DES SIGLES ET ABRÉVIATIONS .....	XV
CHAPITRE 1 INTRODUCTION.....	1
1.1 Contexte .....	1
1.2 Problématique.....	2
1.3 Objectifs .....	3
1.3.1 Question de recherche .....	3
1.3.2 Objectif général .....	3
1.3.3 Objectifs spécifiques .....	3
1.4 Contribution .....	4
CHAPITRE 2 CONCEPTS DE BASE.....	5
2.1 Les circuits intégrés.....	5
2.1.1 Définition .....	5
2.1.2 Circuits intégrés tridimensionnels (3D) .....	5
2.1.3 Problèmes liés à la chaleur dans les circuits tridimensionnels.....	6



2.1.4	Systèmes de refroidissement existants .....	6
2.2	Transfert de chaleur.....	8
2.2.1	Définition de la température.....	8
2.2.2	Les modes de transfert de chaleur .....	8
2.2.3	Conservation d'énergie et bilan thermique .....	9
2.2.4	Les régimes de transfert de chaleur.....	10
CHAPITRE 3	REVUE DE LITTÉRATURE .....	12
3.1	Les modèles thermiques analytiques.....	12
3.2	Les modèles thermiques compacts .....	12
3.3	Les modèles thermiques numériques .....	13
3.3.1	Principe et méthodologies .....	13
3.3.2	Travaux sur la simulation thermique du régime permanent.....	18
3.3.3	Travaux sur la simulation thermique du régime transitoire .....	18
3.3.4	Travaux sur la simulation des régimes transitoire et permanent .....	19
3.4	Comparaison entre les simulateurs existants et ICTherm .....	20
CHAPITRE 4	ICTHERM : CADRE THÉORIQUE POUR LA RÉOLUTION DES ÉQUATIONS .....	22
4.1	Méthodologie générale.....	22
4.2	Méthode de résolution d'équations pour la simulation thermique en régime transitoire... .....	22
4.2.1	Décomposition des opérateurs .....	23
4.2.2	Schéma itératif de correction de l'erreur de décomposition .....	24
4.2.3	Algorithme complet de la méthode proposée.....	24
4.3	Méthode de résolution d'équations pour la simulation thermique en régime permanent .. .....	25

4.3.1	Cas d'un circuit solide.....	26
4.3.2	Cas d'un circuit contenant des composants fluidiques.....	27
4.3.3	Préconditionnement des algorithmes .....	28
4.4	Synthèse .....	28
4.4.1	Validation et résultats.....	29
4.4.2	Résumé des méthodologies .....	29
4.4.3	Limitations .....	30
CHAPITRE 5 ICTHERM : DÉFINITION ET MISE EN PLACE DE L'OUTIL.....		32
5.1	L'outil ICTherm .....	32
5.2	Caractéristiques d'un outil de simulation efficace .....	32
5.3	Défis des circuits futurs.....	33
5.3.1	Circuit à refroidissement microfluidique du projet STREAMS.....	33
5.3.2	Circuit intégrant des composants optiques.....	34
5.4	Définitions des besoins et flot de définition du simulateur ICTherm .....	35
5.5	Le choix du langage de développement .....	36
5.5.1	Python.....	36
5.5.2	Le choix du Python .....	37
5.6	Architecture de l'outil .....	38
5.7	Le module d'entrée.....	39
5.8	La modélisation .....	42
5.8.1	Modélisation des composants et géométries .....	43
5.8.2	Modélisation des matériaux .....	43
5.8.3	Paramètres de simulation .....	44
5.9	Le maillage.....	44

5.10	La discrétisation .....	45
5.11	La résolution.....	47
5.12	Le support de nouvelles options et fonctionnalités .....	47
5.13	Le module de sortie .....	47
CHAPITRE 6 RÉSULTATS EXPÉRIMENTAUX .....		49
6.1	Choix des circuits .....	49
6.2	Intel Atom D2700.....	50
6.2.1	Description du circuit .....	50
6.2.2	Modélisation.....	50
6.2.3	Simulation .....	51
6.3	Circuit intégré tridimensionnel refroidi par écoulement liquide .....	52
6.3.1	Description du circuit .....	53
6.3.2	Modélisation.....	53
6.3.3	Simulation .....	54
6.4	Circuit optique.....	56
6.4.1	Description du circuit .....	56
6.4.2	Modélisation.....	57
6.4.3	Simulation .....	58
6.5	Synthèse des résultats.....	59
CHAPITRE 7 CONCLUSION .....		60
7.1	Synthèse .....	60
7.2	Limitations .....	62
7.3	Améliorations et travaux futurs.....	62
RÉFÉRENCES.....		64

## LISTE DES TABLEAUX

Tableau 3.1: Comparaison entre les méthodes de discrétisation temporelle.....	17
Tableau 3.2: Comparaison entre les simulateurs numériques existants et l’outil ICTherm.....	21
Tableau 6.1: Éléments du modèle thermique de l’Atom D2700.....	51
Tableau 6.2: Éléments du modèle thermique .....	54
Tableau 6.3: Valeurs de température minimale et maximale de la matrice des lasers à différentes valeurs de puissances. ....	59

## LISTE DES FIGURES

Figure 2.1: Structure type d'un circuit intégré [4] .....	5
Figure 2.2: Exemple de circuit intégré tridimensionnel [4] .....	6
Figure 2.3: Dissipateur de chaleur.....	7
Figure 2.4: Principe du refroidissement par écoulement liquide [11].....	7
Figure 2.5: La loi de conservation d'énergie dans un système [13].....	10
Figure 3.1: Les différentes méthodes de discrétisation et résolution .....	13
Figure 3.2: Exemples de maillage [4] .....	15
Figure 4.1: Comparaison entre les températures obtenues par imagerie et par simulation [4] .....	29
Figure 4.2: Méthodologies sélectionnées par rapport à l'état de l'art.....	30
Figure 5.1: Flot de l'outil ICTherm.....	32
Figure 5.2 Circuit à refroidissement microfluidique du projet STREAMS [33] .....	34
Figure 5.3 Circuit intégré tridimensionnel intégrant des composants optiques [34] .....	34
Figure 5.4 Flot de définition et modélisation du simulateur ICTherm.....	36
Figure 5.5: Architecture de l'outil ICTherm .....	38
Figure 5.6: Fenêtre pour ajouter, dupliquer et mettre à jour les géométries du circuit .....	40
Figure 5.7: Fenêtre pour configurer les matériaux dont les composants du circuit sont faits .....	41
Figure 5.8: Fenêtre pour configurer les paramètres de simulation.....	42
Figure 5.9: Un maillage orthogonal structuré conforme [36] .....	44
Figure 5.10: Flux de chaleur échangés dans une cellule [36] .....	46
Figure 5.11: Calcul des capacitances pour une cellule.....	46
Figure 5.12: Carte thermique d'un circuit intégré tridimensionnel obtenu par sa simulation à travers l'outil ICTherm .....	48
Figure 6.1: Plan de masse de l'Atom D2700 [7].....	50

Figure 6.2: Modèle thermique proposé [37].....	51
Figure 6.3: Comparaison des températures moyennes obtenues par simulation du régime permanent de l'Atom D2700 par les simulateurs Comsol et ICTherm pour une activité du processeur à 50%, 90 % et 100% .....	52
Figure 6.4: Circuit multicouche du premier test .....	53
Figure 6.5: Disposition des sources de chaleur [36] .....	53
Figure 6.6: Carte thermique du circuit après simulation du régime transitoire.....	55
Figure 6.7: Comparaison des températures moyennes des quatre puces du circuit lors de la simulation par ICTherm et dans l'implémentation en dur .....	56
Figure 6.8: Architecture du circuit optique [34] .....	57
Figure 6.9: Architecture de la couche contenant les lasers [34].....	57
Figure 6.10: Architecture de la couche contenant les lasers .....	58
Figure 6.11: Distribution de la température dans la matrice des lasers lors de l'activation du laser numéro 13 avec une puissance de 10mW .....	58

## LISTE DES ALGORITHMES

Algorithme 4.1 : Pseudo-code de l'algorithme de Thomas [4] .....	24
Algorithme 4.2: Pseudo-code de l'algorithme de résolution du régime transitoire proposé [4] .....	25
Algorithme 4.3: Pseudo-code de l'algorithme du gradient conjugué [4] .....	26
Algorithme 4.4: Pseudo-code de l'algorithme GMRES [4] .....	27
Algorithme 4.5: Pseudo-code de l'algorithme de décomposition incomplète ILU [4] .....	28

## LISTE DES SIGLES ET ABRÉVIATIONS

3D	Three-dimensional
ADI	Altenating Direct Implicit
C4	Controlled Collapse Chip Connection
CG	Conjugate Gradient
CMOS	Complementary Metal Oxyde Semiconductor
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
GMRES	Generalised Minimal Residual Method
GPU	Graphics Processing Unit
ILU	Incomplete Lower Upper
LU	Lower Upper
ONI	Optical Network Interface
SPO	Split Operator
TIM	Thermal Interface Material



## CHAPITRE 1 INTRODUCTION

Ce chapitre présente la problématique de dissipation de chaleur dans les circuits intégrés ainsi que l'importance de la simulation thermique et ses défis pour enfin énoncer les objectifs et la contribution de ce travail.

### 1.1 Contexte

Depuis leur apparition en 1958, les circuits intégrés connaissent une évolution exponentielle. Les premiers circuits intégrés à leur début étaient formés par quelques transistors pour atteindre aujourd'hui des milliards de transistors et de connexions métalliques qui dégagent de la chaleur sur une surface réduite. Le flux de chaleur élevé généré induit une augmentation de la température du circuit intégré. Cette élévation implique la dégradation des performances et des fonctionnalités et la diminution de la durée de vie de ces circuits [1].

Avec l'introduction des circuits intégrés tridimensionnels (3D) [2], le défi de garder ces circuits à une température acceptable devient plus ardu. Bien que ces circuits présentent un grand nombre d'avantages comme une taille compacte, une grande bande passante et un grand nombre de fonctionnalités, ils introduisent de nouveaux défis de conception en particulier pour le contrôle efficace de leur température [3]. En effet, les circuits intégrés tridimensionnels présentent une structure multicouche ; cette superposition verticale de couches comportant un ensemble de sous-composants, empêche une dissipation efficace de la chaleur générée au niveau des transistors et des connexions métalliques. Face à ce problème de surréchauffement, plusieurs méthodes de refroidissement combinées à des techniques de réduction de consommation des puissances ont été adoptées. Les diffuseurs de chaleur, les dissipateurs de chaleur passifs et actifs et le refroidissement par écoulement d'un fluide constituent les techniques de refroidissement les plus répandues [4] [5].

Une conception efficace de tels circuits nécessite des compétences multidisciplinaires, ce qui constitue un grand défi pour les concepteurs. Afin d'explorer les différentes possibilités de modélisation de leurs circuits et de valider l'efficacité de leur conception du point de vue thermique, les concepteurs ont besoin d'outils leur permettant de quantifier la chaleur dégagée et d'évaluer de multiples alternatives de conception sans le surcôt en temps et en argent amené par le prototypage. Pour ce faire, les concepteurs ont recours à la simulation thermique, qui à partir

d'une description détaillée de leur circuit et de l'énergie dégagée leur fournit une description des échanges thermiques pendant une durée de temps choisie.

## 1.2 Problématique

Les circuits intégrés tridimensionnels sont formés de plusieurs couches contenant un nombre de composants électroniques fabriqués avec des matériaux différents et possédant ainsi des propriétés thermiques différentes. L'adoption de nouvelles méthodes de refroidissement comme le refroidissement par écoulement d'un fluide [6] accentue l'hétérogénéité caractérisant ces circuits en introduisant des matériaux liquides ou gazeux. La simulation des échanges de chaleur dans ce type de circuits doit tenir compte de cet aspect multi disciplinaire et de cette hétérogénéité.

Les circuits intégrés tridimensionnels sont très compacts et présentent des géométries complexes, réduites et non uniformes sur les trois dimensions ; allant de quelques milliers de micromètres à quelques dizaines. Une simulation thermique efficace et précise doit prendre en considération des contraintes structurelles et dimensionnelles de ces circuits.

Certaines couches des circuits intégrés tridimensionnelles contiennent plusieurs sources de chaleurs, de dimensions diverses et générant des puissances variables ; ce qui complique l'évaluation des flux thermiques échangés.

Pour bien évaluer sa conception, le concepteur doit pouvoir étudier les valeurs des températures finales atteintes après stabilisation ainsi que l'évolution de celle-ci dans le temps. La simulation thermique doit donc couvrir les deux régimes permanent et transitoire.

Les circuits intégrés sont en permanente évolution, on parle aujourd'hui de circuits optiques en ajoutant aux couches électroniques des couches formées par des lasers optiques ; de nouveaux types de circuits refroidis par écoulement de fluides conçus de manière à faire varier certains paramètres (comme le débit d'écoulement du fluide) en cours de fonctionnement pour mieux réguler la température. Une bonne simulation thermique doit prendre en considération ces nouvelles fonctionnalités introduites par les circuits du futur.

Pour résumer, la simulation thermique doit prendre en considération :

- L'hétérogénéité des composants du circuit.
- Les géométries complexes, tridimensionnelles du circuit intégré.

- Les sources de chaleur multiples, de dimensions différentes et générant des puissances variables.
- Les régimes permanent et transitoire.
- Les fonctionnalités évolutives des circuits intégrés.

## 1.3 Objectifs

Cette partie énonce la question de recherche et les objectifs généraux et spécifiques de ce travail.

### 1.3.1 Question de recherche

Ce projet répond à la question de recherche suivante :

*Comment réaliser un simulateur thermique générique permettant la simulation des échanges thermiques dans les circuits intégrés tridimensionnels?*

### 1.3.2 Objectif général

L'objectif général de ce mémoire est d'implémenter un simulateur thermique fiable, précis et portable pour tous types de circuits intégrés tridimensionnels allant des circuits classiques jusqu'aux circuits de nouvelles générations.

### 1.3.3 Objectifs spécifiques

Les objectifs spécifiques de ce projet sont :

- Modélisation générique des circuits intégrés.
- Discrétisation du circuit.
- Implémentation de la simulation pour les régimes permanent et transitoire.
- Support des nouvelles fonctionnalités des futurs circuits intégrés tridimensionnels.
- Améliorer l'ergonomie de l'outil : faciliter l'utilisation de l'outil à travers une interface utilisateur pour normaliser et accélérer la saisie des entrées et ajouter une représentation graphique des résultats.

## 1.4 Contribution

Notre travail s'inscrit dans le cadre d'une continuité de travaux sur la simulation thermique des circuits intégrés tridimensionnels réalisés dans le groupe HESL (Heterogeneous Embedded Systems Laboratory). Nous considérons comme point de départ les résultats des travaux effectués dans notre laboratoire de recherche par monsieur Alain Fourmigue [4]. Ses travaux proposent des solveurs efficaces pour la simulation thermique, et leur application sur quelques études de cas concrètes.

En exploitant ces solveurs, les contributions proposées sont :

1. L'implémentation d'un outil complet de simulation thermique permettant une modélisation simple du circuit en entrée, une simulation rapide, fiable et paramétrable pour les régimes transitoire et permanent.
2. L'adoption de solveurs pour assister les concepteurs des futurs circuits intégrés en supportant de nouvelles fonctionnalités et permettre un ajout et une mise à jour simple de nouvelles caractéristiques.
3. L'application de l'outil sur trois circuits illustratifs :
  - Un circuit représentant le processeur AtomD2700 [7].
  - Un circuit microfluidique basé sur le processeur SPARC T1 [8].
  - Un circuit intégrant des composants optiques.

## CHAPITRE 2 CONCEPTS DE BASE

Dans ce chapitre, nous allons détailler quelques concepts de base liés au sujet de recherche. Nous allons introduire les circuits intégrés et leurs spécificités et énoncer les concepts liés au phénomène de transfert de chaleur.

### 2.1 Les circuits intégrés

#### 2.1.1 Définition

Un circuit intégré est un composant électronique assurant un ensemble de fonctionnalités plus ou moins complexes. Il peut être formé de plusieurs sous composants.

Le premier circuit intégré a été inventé en 1958 par Jack Kilby, il était alors composé d'un ensemble de transistors liés à la main ; il a depuis connu une évolution exponentielle pour atteindre des milliards de transistors assurant des fonctionnalités de grande complexité.

Le circuit intégré est formé par un boîtier d'encapsulation incluant une ou plusieurs puces. Le boîtier d'encapsulation assure la protection du circuit, sa connexion avec la carte mère et contribue au refroidissement du circuit. La puce est le composant de base qui contient la fonctionnalité du circuit [9]. La Figure 2.1 illustre une structure type d'un circuit intégré.

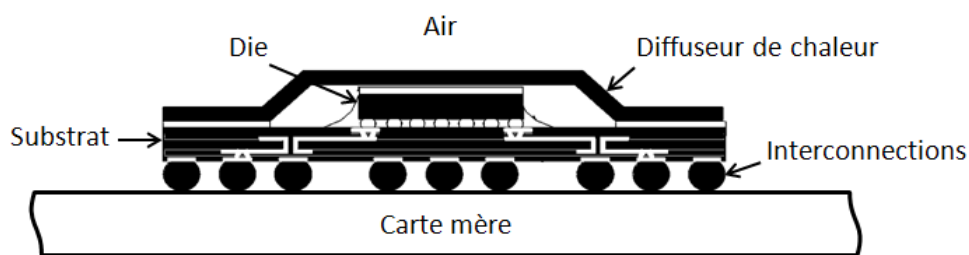


Figure 2.1: Structure type d'un circuit intégré [4]

#### 2.1.2 Circuits intégrés tridimensionnels (3D)

Depuis leur apparition, les circuits intégrés connaissent une évolution continue afin d'assurer plus de fonctionnalités et plus de compacité. Dans ce cadre, de nouveaux circuits sont apparus amenant une nouvelle topologie en couches. Les circuits intégrés tridimensionnels se composent donc de plusieurs couches connectées par des TSV (Through Silicon Via) [10], il s'agit d'interconnecteurs électriques verticaux de haute performance. Cette technologie offre plusieurs

avantages comme une taille réduite, un nombre de composants et fonctionnalités élargis, une bande passante plus large et diverses possibilités de conception.

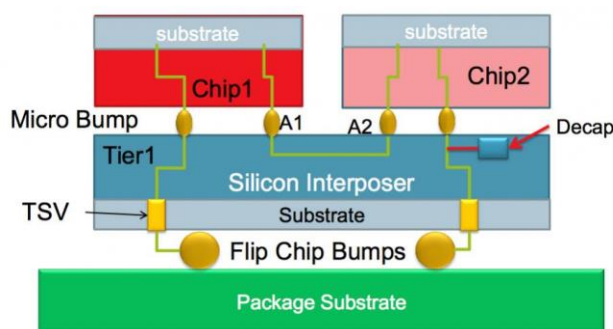


Figure 2.2: Exemple de circuit intégré tridimensionnel [4]

### 2.1.3 Problèmes liés à la chaleur dans les circuits tridimensionnels

Les circuits intégrés dégagent de la chaleur au niveau des transistors et des connexions métalliques ; en particulier pour les circuits tridimensionnels dont l'architecture en couches empêche une dissipation efficace de la chaleur produite. Cette élévation de température est particulièrement nocive pour le bon fonctionnement du circuit et ses performances, sa durée de vie ; un surréchauffement continu induit une destruction des composants de celui-ci.

### 2.1.4 Systèmes de refroidissement existants

Le réchauffement des circuits intégrés a un effet très néfaste, maintenir le circuit à une température acceptable est primordial lors de la phase de conception. Pour remédier à ce problème, plusieurs systèmes de refroidissement ont été mis en place [4] [5].

#### Diffuseurs de chaleur

Les diffuseurs de chaleurs sont des plaques d'aluminium ou de cuivre ajouté au circuit ou intégré dans son boîtier d'encapsulation. La diffusion de chaleur est assurée par leur haute conductivité thermique et leur surface importante par rapport à la taille du circuit.

#### Dissipateurs de chaleur

Les dissipateurs de chaleurs sont des dispositifs souvent munis d'ailettes, ils contribuent à évacuer la chaleur du circuit vers l'air ambiant à travers le phénomène de convection. Lorsque cette convection est faite naturellement, on parle de dissipateurs passifs. Les dissipateurs actifs essayent

de provoquer le phénomène de convection en intégrant des ventilateurs aux ailettes. Cette convection, dite alors convection forcée, assure plus de refroidissement que la dissipation passive.

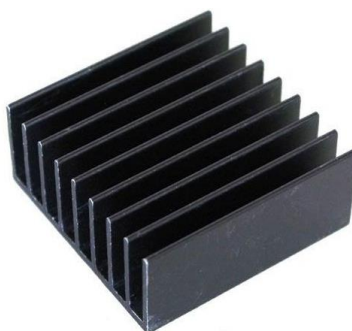


Figure 2.3: Dissipateur de chaleur

### Refroidissement par écoulement liquide

Le refroidissement par écoulement d'un fluide est généralement employé dans les circuits intégrés tridimensionnels où les techniques de dissipation et diffusion de chaleur peinent à les garder à une température raisonnable. Des couches où de microcanaux sont creusés, sont intégrées au circuit pour permettre l'écoulement du fluide. La température du fluide inférieure à celle du système et sa vitesse d'écoulement permettent alors le refroidissement de celui-ci. Cette technique est assez efficace, mais contribue à ajouter de la complexité à la conception de ce type de circuit ; en effet, les dimensions des microcanaux, le type du fluide, la vitesse d'écoulement sont des paramètres additionnels que le concepteur des circuits intégrés tridimensionnels doit considérer.

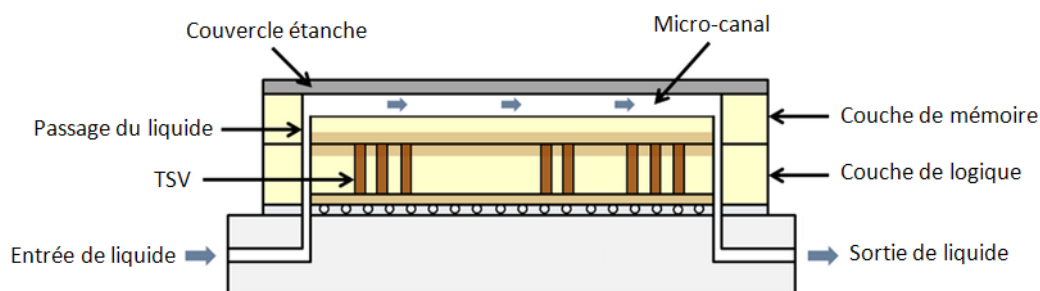


Figure 2.4: Principe du refroidissement par écoulement liquide [11]

La Figure 2.4 illustre le fonctionnement de la technique de refroidissement par écoulement fluide. Le liquide pompé circule à travers des microcanaux pour permettre de refroidir le circuit.

## 2.2 Transfert de chaleur

### 2.2.1 Définition de la température

La température est une grandeur physique liée à la sensation du froid et du chaud. Elle peut être définie de plusieurs façons ; en physique, elle traduit à l'échelle macroscopique l'agitation cinétique des particules microscopiques qui constituent la matière ; en thermodynamique, elle est définie comme l'inverse du taux de variation de l'entropie d'un système par son énergie. La température peut être mesurée en degrés Celsius, Fahrenheit ou Kelvin.

### 2.2.2 Les modes de transfert de chaleur

Ils existent trois modes de transfert de chaleur : par conduction, par convection et par rayonnement [12] [13].

#### Transfert de chaleur par conduction

La conduction est le transfert de chaleur dans un milieu stationnaire. Il s'agit du transfert de chaleur dans un solide ou un fluide au repos résultant d'un échange d'énergie entre les particules. Elle est calculée par la loi de Fourier [12] :

$$\overrightarrow{q_{cond}} = -k\overrightarrow{\nabla T}S \quad (2.1)$$

$\overrightarrow{q_{cond}}$  étant le flux de chaleur conductif circulant à travers la surface  $S$ . Le coefficient  $k$  est le coefficient de conductivité thermique, il s'agit d'une propriété du matériau dont le solide ou le fluide est fait, il est exprimé en  $W/m.K$ .  $S$  est la surface d'échange.

Dans un circuit intégré, le transfert par conduction se fait souvent entre les composants solides du circuit.

#### Transfert de chaleur par convection

Le phénomène de convection est généralement associé à un échange de chaleur entre matériaux solides et matériaux liquides ou gazeux en mouvement. Cette convection peut être naturelle ou



forcée selon la présence ou non de facteur externe (pompe, ventilateur...) qui cause un différentiel de pression. L'écoulement du fluide peut être ordonné dit laminaire ou rotationnel désordonné dit turbulent. Le niveau de turbulence dépend de la vitesse de l'écoulement. La convection est calculée par la loi de Newton [12] :

$$\overrightarrow{q_{conv}} = h(T_{surface} - T_{fluide})S\vec{n}_{surface} \quad (2.2)$$

Où  $\overrightarrow{q_{conv}}$  est le flux thermique par convection ou flux convectif,  $h$  est le coefficient de convection ou coefficient de transfert thermique exprimé en  $W/m^2K$ .  $S$  est la surface d'échange.  $T$  est la température.

Dans un circuit intégré, une convection naturelle peut avoir lieu entre les couches limites et l'air ambiant ; une convection forcée si un dissipateur actif est utilisé ou à travers les parois des microcanaux lors d'un refroidissement par écoulement fluide.

### Transfert de chaleur par rayonnement

Le rayonnement est un type de transfert de chaleur complètement différent de la convection et la conduction puisqu'il n'y a pas d'échanges de chaleur entre deux corps adjacents. En effet indépendamment de sa température un corps émet une onde électromagnétique vers son environnement. Le rayonnement est mesuré par la loi de Stefan-Boltzman [12] :

$$\overrightarrow{q_{rad}} = \sigma(T_{surface}^4 - T_{air}^4)S\vec{n}_{surface} \quad (2.3)$$

Où  $\overrightarrow{q_{rad}}$  est le flux de chaleur par rayonnement exprimé en Watts ;  $T_s$  est la température de la surface ;  $\sigma$  est la constante de Stefan-Boltzmann qui est égale à  $5.67 E -8 \text{ Watts/m}^2 K^4$ .

### 2.2.3 Conservation d'énergie et bilan thermique

Pour un système, la loi de conservation d'énergie connue comme la première loi de la thermodynamique est définie par l'équation 2.4 qui est illustrée par la Figure 2.5:

$$\dot{E}_{in} - \dot{E}_{out} + \dot{E}_g = \dot{E}_{st} \quad (2.4)$$

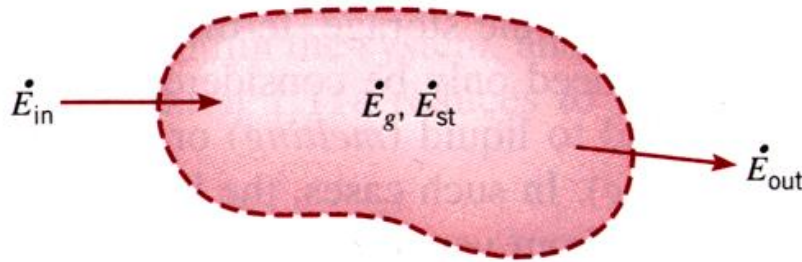


Figure 2.5: La loi de conservation d'énergie dans un système [13]

$\dot{E}_{in}$  étant la quantité d'énergie entrante,  $\dot{E}_{out}$  la quantité d'énergie sortante,  $\dot{E}_{st}$  l'énergie emmagasinée et  $\dot{E}_g$  l'énergie générée.

L'application directe de cette loi de conservation d'énergie donne l'équation de bilan thermique d'un système. La résolution de l'équation de bilan thermique donnera la distribution thermique au sein d'un circuit. Le bilan thermique sur un volume  $V$  peut-être écrit sous la forme :

$$\int_V c_v \frac{\partial T}{\partial t} dV = \int_S \vec{q} \cdot \vec{ds} + \int_V P_v dV \quad (2.5)$$

Où  $c_v$  constitue la capacité volumique thermique du matériau,  $T$  est la température du volume  $V$ ,  $\vec{q}$  est le flux de chaleur surfacique reçue par le volume  $V$  à travers la surface  $S$ , ce flux peut-être conductif, convectif ou par rayonnement et  $P_v$  est la puissance volumique dissipée dans le volume

Cette équation appelée équation de chaleur est une équation aux dérivées partielles dont la résolution donnera la distribution de la chaleur dans chaque point du système.

## 2.2.4 Les régimes de transfert de chaleur

### Le régime permanent

Le régime permanent est le régime d'un système en état stable. Il intervient lorsque le comportement du système n'évolue plus. Dans ce régime, la distribution de température n'étant plus sujette à évolution est indépendante du temps [13].

### Le régime transitoire

On parle de régime transitoire lorsque le système n'a pas atteint l'état stable. La distribution de la température encore en évolution est alors une fonction du temps [13]. Le temps d'amortissement dépend de la composition du circuit : les caractéristiques des matériaux dont il est fait et de son boîtier d'encapsulation. Le temps pour atteindre la stabilité peut varier entre quelques secondes et une minute pour les circuits intégrés classiques [14].

## **CHAPITRE 3    REVUE DE LITTÉRATURE**

Plusieurs travaux de recherche ont abordé le problème de la modélisation et simulation thermique des circuits intégrés tridimensionnels. De ces travaux, plusieurs modèles de résolution de l'équation de chaleur ont vu le jour. Ces modèles peuvent être classifiés en trois catégories : le modèle thermique analytique, le modèle thermique compact et le modèle thermique numérique.

Dans ce chapitre, nous allons d'abord introduire brièvement les deux premiers modèles ainsi que les simulateurs existants basés sur ces modèles. Par la suite, nous allons discuter plus en détail le modèle thermique numérique qui est à la base de nos travaux de recherche.

### **3.1 Les modèles thermiques analytiques**

Les modèles analytiques se basent sur une résolution mathématique de l'équation de chaleur [12]. Ces modèles sont très précis, mais sont très lourds à mettre en place et ne sont applicables que dans le cas d'un régime permanent.

Dans la littérature, quelques outils se sont appuyés sur un modèle analytique :

1. Le logiciel THERMAN [15] basé sur les séries de Fourier, il permet la modélisation de circuit avec quelques couches uniformes.
2. Le logiciel NUMANA [16] est basé sur une discrétisation numérique et une résolution analytique par les fonctions de Green : il permet de modéliser et simuler des circuits avec des composants fabriqués dans des matériaux hétérogènes pour le régime permanent.

### **3.2 Les modèles thermiques compacts**

Les modèles compacts reproduisent le comportement du circuit en créant comme leur nom l'indique un modèle simplifié de celui-ci qui n'est pas fidèle à la structure et la composition du circuit, mais sert à reproduire son comportement thermique [17]. Les modèles compacts sont très rapides, mais peu précis puisqu'ils se basent sur une évaluation de la température en seulement quelques points du circuit [17].

Certains modèles dans la littérature sont basés sur des modèles compacts :

1. Pour DELPHI [18], le modèle compact du circuit à simuler est modélisé par des résistances thermiques, des jonctions et des broches. La première version du modèle DELPHI ne supporte que

le régime permanent, mais une extension qui permet d'effectuer de très lentes simulations du régime transitoire a été réalisée.

2. Hotspot [19] modélise le circuit à un niveau architectural simple, chaque composant étant représenté par un nœud dans le modèle compact du circuit. Hotspot supporte le régime transitoire et permanent. De nouvelles versions de Hotspot ont été développées pour supporter les circuits tridimensionnels et pour apporter plus de précision.

### 3.3 Les modèles thermiques numériques

Dans cette section, nous allons présenter les modèles thermiques numériques et énumérer les méthodes existantes pour la discrétisation et la résolution de ce type de modèle.

#### 3.3.1 Principe et méthodologies

Dans les modèles thermiques numériques, la résolution de l'équation de la chaleur est ramenée à une résolution d'un problème d'algèbre linéaire. Cette transformation est assurée à travers une discrétisation spatiale et temporelle.

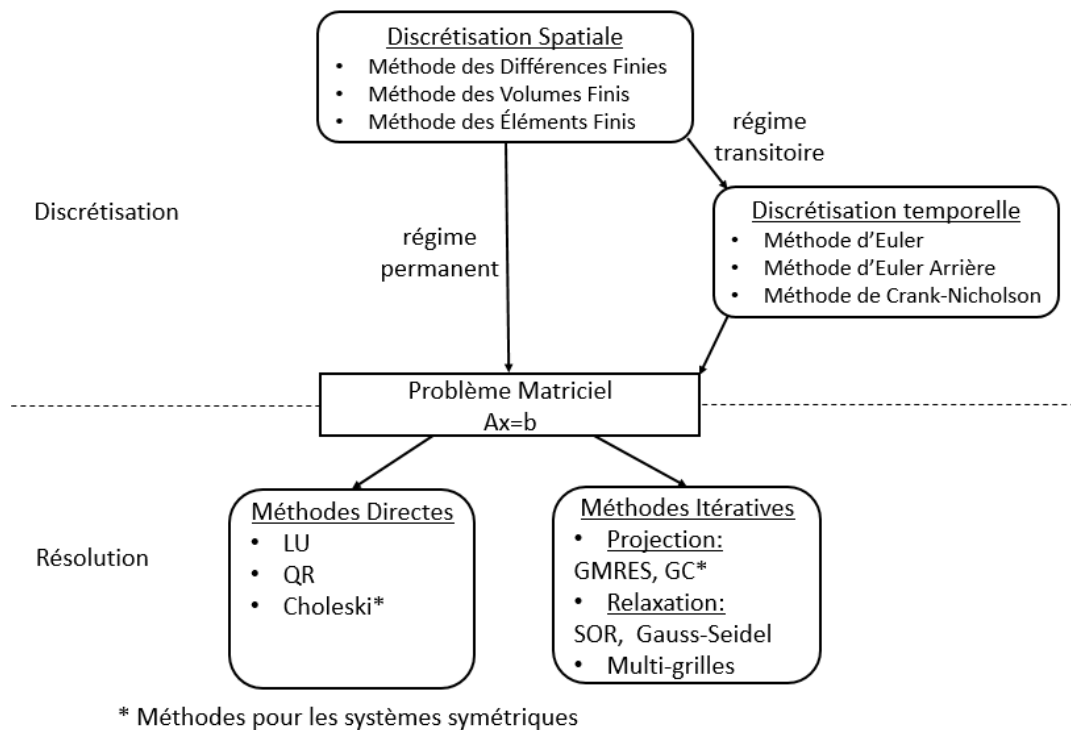


Figure 3.1: Les différentes méthodes de discrétisation et résolution

La Figure 3.1 illustre les différentes méthodes de discrétisation spatiale et temporelle et les méthodes de résolution du système d'équations résultant de cette discrétisation. Ces méthodes seront détaillées dans les sections suivantes.

### **Discrétisation spatiale**

Trois principales méthodes sont connues dans la discrétisation spatiale des équations aux dérivées partielles [12].

#### Méthode des différences finies

C'est la méthode la plus ancienne et la plus simple de discrétisation spatiale, elle consiste en une recherche de solutions approximatives d'équations aux différences partielles à l'aide des séries de Taylor.

#### Méthode des volumes finis

La méthode des volumes finis (FVM) est adaptée aux géométries complexes. Comme la méthode des différences finies, elle permet de donner des solutions approchées d'équations aux dérivées partielles, mais grâce à des approximations d'intégrales. La méthode FVM travaille avec des volumes de contrôle définis à partir du maillage et considère les valeurs moyennes des grandeurs physiques dans ces volumes.

#### Méthode des éléments finis

La méthode des éléments finis permet de traiter les problèmes plus généraux que ceux limités à la loi de conservation. Elle est donc plus générale que les méthodes des volumes finis et différences finies, mais plus complexe à mettre en place. Elle recherche les approximations des solutions à partir de fonctions d'interpolation.

#### Synthèse des méthodologies

Toutes les méthodes de discrétisation par rapport à la variable de l'espace des systèmes tridimensionnels opèrent sur un maillage. Un maillage est une représentation en cellules du circuit. Les cellules peuvent être de différentes formes comme illustrées dans la Figure 3.2.

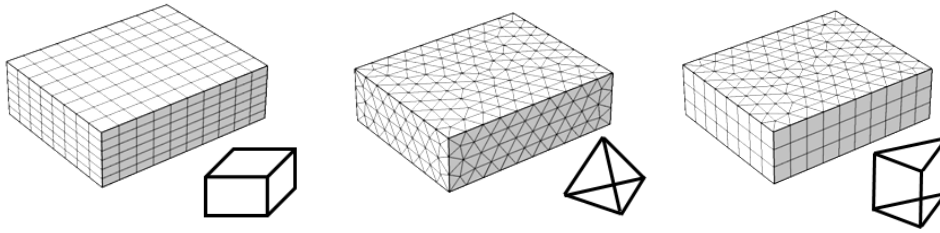


Figure 3.2: Exemples de maillage [4]

Une fois discrétisée, quelle que soit la méthode utilisée, l'équation de chaleur se présente sous la forme :

$$C\dot{T} + GT = P \quad (3.1)$$

Où  $G$  est la matrice de conductance,  $C$  la matrice de capacitance,  $P$  le vecteur de puissance et  $T$  est le vecteur de température. Le terme  $C\dot{T}$  est nul lorsqu'il s'agit d'un régime permanent, puisqu'il est indépendant de la variable temps.

Les méthodes des différences finies et les volumes finis restent plus simples à utiliser pour discrétiser le problème de transfert de chaleur que la méthode des éléments finis.

### Discrétisation temporelle

Dans le cas d'un régime transitoire, après discrétisation spatiale, l'équation de la chaleur est sous la forme :

$$C\dot{T} + GT = P \quad (3.1)$$

Pour ramener cette équation à une équation matricielle de premier ordre sous la forme  $Ax=b$ , une discrétisation par rapport à la variable de temps est nécessaire. Le principe de cette discrétisation temporelle est de diviser le temps continu en des intervalles de temps discret. À chaque pas de temps, la simulation est reconduite et la nouvelle température est calculée à partir de la température au temps précédent, calculée lors de l'itération passée.

Plusieurs méthodes pour assurer cette discrétisation existent [12] : la méthode d'Euler classique, la méthode d'Euler arrière et la méthode de Crank-Nicholson.

### La méthode d'Euler

La méthode d'Euler est une méthode de discrétisation avant. L'équation 3.1 est ramenée à :

$$C \frac{(T_{m+1} - T_m)}{\Delta t} + G T_m = P_m \quad (3.2)$$

$T_{m+1}$  et  $T_m$  étant respectivement les températures au pas numéro  $m+1$  et  $m$ .  $C$  est la matrice capacitance,  $G$  la matrice conductance et  $P_m$  le vecteur puissance au pas de simulation  $m$ .

Mise sous la forme  $Ax=b$ , on obtient :

$$T_{m+1} = (I + \Delta t C^{-1} G) T_m + \Delta t C^{-1} P_m \quad (3.3)$$

#### La méthode d'Euler arrière

La méthode d'Euler arrière ramène l'équation 3.1 à :

$$C \frac{(T_m - T_{m-1})}{\Delta t} + G T_m = P_{m-1} \quad (3.4)$$

Mise sous la forme  $Ax=b$ , on obtient :

$$(I + \Delta t C^{-1} G) T_{m+1} = T_m + \Delta t C^{-1} P_m \quad (3.5)$$

#### La méthode de Crank-Nicholson

La méthode de Crank-Nicholson est une méthode de discrétisation centrée. L'équation 3.1 est ramenée à :

$$C \frac{(T_{m+1} - T_m)}{\Delta t} + G \frac{(T_{m+1} + T_m)}{2} = P_m \quad (3.6)$$

Mise sous la forme  $Ax=b$ , on obtient :

$$\left( I + \frac{\Delta t}{2} C^{-1} G \right) T_{m+1} = \left( I - \frac{\Delta t}{2} C^{-1} G \right) T_m + \Delta t C^{-1} P_m \quad (3.7)$$

#### Synthèse des méthodologies

Une comparaison entre les différentes méthodes de discrétisation temporelle selon les critères du cout, de la complexité et de la stabilité de l'algorithme est donnée dans le Tableau 3.1.



Tableau 3.1: Comparaison entre les méthodes de discrétisation temporelle

<i>Méthode</i>	<i>Cout</i>	<i>Complexité</i>	<i>Stabilité</i>
<i>Euler</i>	<i>faible</i>	$O(n)$	<i>conditionnellement stable : divergence lors de grand pas de simulation</i>
<i>Euler arrière</i>	<i>élevé : cout d'inversion de la matrice <math>(I + \Delta t</math> <math>C^{-1}G)</math></i>	$O(n^3)$ : pour des matrices denses	<i>inconditionnellement stable : Précision acceptable même pour les grands pas de simulation</i>
<i>Crank-nicholson</i>	<i>élevé : cout d'inversion de la matrice <math>(I + \frac{\Delta t}{2}</math> <math>C^{-1}G)</math></i>	$O(n^3)$ : pour des matrices denses	<i>théoriquement inconditionnellement stable, mais en pratique des oscillations dans les valeurs température obtenues pour un pas <math>\Delta t &gt; 10ms</math>.</i>

## Résolution

Les modèles basés sur une méthode de résolution numérique sont plus flexibles et permettent une résolution plus efficace et précise de l'équation de chaleur que les modèles analytiques et compacts. Le système matriciel obtenu suite à la phase de discrétisation peut être résolu par des méthodes directes ou des méthodes itératives. Le but de ces méthodes est de résoudre le système sous forme  $Ax=b$  obtenu par l'étape de discrétisation.

### Méthodes directes

Les méthodes directes permettent une résolution en un nombre fixe d'étapes. Elles sont connues pour leur précision, mais ne sont adéquates que pour les systèmes de taille réduite ou moyenne. Plusieurs méthodes directes sont utilisées dans l'algèbre linéaire comme la décomposition LU, la décomposition QR et la méthode de Choleski [20].

### Méthodes itératives

Les méthodes itératives se basent, comme leur nom l'indique sur des résolutions sur plusieurs itérations. À chaque itération, une solution approchée est donnée. Les calculs s'arrêtent lorsque des critères de convergence préalablement définis sont atteints. Les méthodes itératives sont performantes pour la résolution de systèmes de grande taille.

#### **3.3.2 Travaux sur la simulation thermique du régime permanent**

Pour des problèmes de grande taille, la simulation du régime permanent basée sur des méthodes itératives est plus efficace que celle basée sur des méthodes directes.

Dans leurs travaux, Mizunuma et al. [21] utilisent la méthode Gauss Seidel ; les résultats montrent que le temps de simulation demandé est très important : pour un système de taille=200, 11 secondes de calcul sont nécessaires. Ces résultats sont dus à la lenteur de convergence de la méthode de Gauss Seidel.

Pour calculer la température dans des circuits composés purement de matériaux solides, Goplen et al. optent pour une discrétisation à l'aide de la méthode des éléments finis avec une résolution avec l'algorithme de gradient conjugué très connu pour son efficacité pour les systèmes matriciels symétriques définis positifs [22]. Goplen et al. ne rapportent pas les temps de simulations obtenus. Les travaux de Feng et al. se basent sur une résolution des systèmes refroidis par écoulement liquide par l'algorithme du gradient conjugué qui est habituellement utilisé pour les calculs faits sur des matrices symétriques. Tout comme Goplen et al., ils ne rapportent pas les temps de simulations obtenus.

#### **3.3.3 Travaux sur la simulation thermique du régime transitoire**

Liu et al. [23] proposent une résolution basée sur l'algorithme itératif Généralisation de la Méthode de Minimisation du Résidu GMRES [24] pour le régime transitoire. Malheureusement, en adoptant cette approche, dans le cas de régime transitoire le calcul est itéré pour chaque pas de simulation et les méthodes itératives ne montrent pas d'efficacité particulière. Toutefois, l'implémentation sur GPU améliore les performances de la méthode Liu et al. Cependant, le temps de simulation reste encore inacceptable pour effectuer des simulations complexes  $864 \text{ sec}$  pour  $1600$  pas de simulation pour un système matriciel de taille  $N=1.7 \times 10^5$ .

Les travaux de Wang et al. [25] se basent sur la méthode ADI (venant de l'anglais Alternating Direction Implicit), le système d'équations est décomposé suivant les trois dimensions spatiales  $x$ ,  $y$  et  $z$  pour obtenir des systèmes tridiagonaux qui sont résolus par l'algorithme de Thomas. Cette Méthode offre de très bons résultats, mais présente deux contraintes majeures : la décomposition tridimensionnelle ne peut être réalisée que sur un maillage orthogonal structuré et les pas de simulation ne doivent pas dépasser les  $10ms$ .

### 3.3.4 Travaux sur la simulation des régimes transitoire et permanent

#### Modèles basés sur des méthodes directes

L'outil de simulation 3D-ICE [26] est développé à l'École Polytechnique Fédérale de Lausanne, il permet la simulation des régimes transitoire et permanent, pour les circuits solides et ceux refroidis par écoulement d'un liquide.

3D-ICE est développé en C, il est basé sur la méthode de résolution directe décomposition LU [27] pour résoudre le système des équations de chaleurs. La décomposition ou factorisation LU permet de décomposer la matrice  $A$  du système  $Ax=b$  à résoudre en un produit d'une matrice triangulaire supérieure et d'une matrice triangulaire inférieure. 3D-ICE utilise la librairie performante SuperLU avec une complexité nettement inférieure à celle de la décomposition LU classique  $O(N^{1.6})$  au lieu de  $O(N^3)$  ; pour des problèmes de taille  $= 10^5$ , la résolution prend aux alentours de 40 secondes. Il a été validé avec l'outil commercial ANSYS et offre une erreur de précision de 4%.

L'utilisation de la librairie SuperLU constitue l'atout principal de 3D-ICE, mais aussi son inconvénient majeur puisque la résolution du système reste assez complexe, lente à l'exécution et gourmande en mémoire. Pour remédier à ce problème, Vincenzi et al. implémentent sur GPU une nouvelle version de 3D-ICE basée sur les réseaux de neurones [28], ce qui accélère la résolution, mais nécessite l'entraînement des réseaux de neurones à chaque changement et ne résout pas la consommation mémoire conséquente de l'outil.

Tout comme 3D-ICE, le simulateur 3D-ACME [29] se base sur la méthode directe de décomposition LU pour résoudre le système d'équations obtenu après discrétisation. Il utilise par contre la librairie KLU qui est légèrement plus performante que SuperLU ; en effet, un système de taille  $N=10^5$  peut être résolu en 30 secondes. Il partage avec 3D-ICE les mêmes limites liées à la

lenteur de l'exécution et la consommation mémoire, et l'incapacité à résoudre des problèmes de taille supérieure à  $10^5$ .

### **Modèles basés sur des méthodes itératives**

Dans leurs travaux, Li et al. [30] présentent un modèle numérique basé sur la méthode multigrilles combinée à une méthode de relaxation pour accélérer la convergence de celle-ci. Ils rapportent de très bons résultats : pour un système de taille  $= 784 \times 10^3$ , seulement 2.3 secondes de temps de calcul sont nécessaires. Mais le bon choix de la combinaison des grilles est une condition pour garantir la qualité des résultats.

### **Simulateurs commerciaux**

Comsol Multiphysics [31] est un logiciel de simulation numérique très populaire, il est basé sur une discrétisation avec la méthode des éléments finis. Il comporte plusieurs modules avec un module transfert de chaleur dédié à la simulation thermique. Le module thermique est très efficace, la modélisation du circuit est simplifiée à l'aide de représentation graphique, mais l'outil est gourmand en mémoire et payant.

Tout comme Comsol Multiphysics, ANSYS [32] est un logiciel de simulation numérique basé sur la méthode des éléments finis. Un module transfert de chaleur est disponible sur l'environnement ANSYS Workbench. La modélisation de circuits intégrés tridimensionnels complexes est compliquée sur ANSYS.

## **3.4 Comparaison entre les simulateurs existants et ICTherm**

Le Tableau 3.2 représente une comparaison entre les simulateurs existants dans la littérature et l'outil ICTherm présenté dans nos travaux. La comparaison est réalisée selon des classes de critères : le régime simulé offert (transitoire et/ou permanent), l'efficacité qui englobe la précision des résultats de simulation et la capacité à résoudre des circuits complexes qui une fois discrétisés donnent des systèmes à résoudre, la gratuité de l'outil et son aptitude à modéliser et supporter les nouvelles fonctionnalités des circuits futurs qui vont être détaillées dans la section 5.3.

Tableau 3.2: Comparaison entre les simulateurs numériques existants et l'outil ICTherm.

<i>Outil</i>	<i>Régime simulé</i>		<i>Efficacité</i>		<i>Gratuité de l'outil</i>	<i>Simulations des circuits futurs</i>
	<i>Transitoire</i>	<i>Permanent</i>	<i>Précision</i>	<i>Résolution des systèmes de grande taille</i>		<i>Support de nouvelles technologies</i>
<i>3D-ICE</i>	✓	✓	✓	✗	✓	✗
<i>3D-ACME</i>	✓	✓	✓	✗	✓	✗
<i>Comsol</i>	✓	✓	✓	✓	✗	✗
<i>ANSYS</i>	✓	✓	✓	✓	✗	✗
<u><i>ICTherm</i></u>	✓	✓	✓	✓	✓	✓

## **CHAPITRE 4    ICTHERM : CADRE THÉORIQUE POUR LA RÉSOLUTION DES ÉQUATIONS**

Ce chapitre décrit les travaux de recherche effectués dans notre laboratoire [4] et qui constituent la base théorique de notre travail. Nous allons détailler les méthodologies et les algorithmes sur lesquels nous nous sommes basés pour implémenter les solveurs utilisés par le simulateur thermique ICTherm.

### **4.1 Méthodologie générale**

Dans le cadre de notre projet, la simulation thermique des circuits intégrés tridimensionnels est basée sur un modèle thermique numérique. Dans un modèle numérique, la résolution de l'équation de chaleur est ramenée à la résolution d'un système d'équations linéaires. Pour ce faire, une phase de discrétisation est nécessaire pour obtenir le système d'équations linéaires sous la forme  $Ax=b$  ou  $x$  est le vecteur température à calculer. Une fois le système matriciel obtenu, sa résolution est assurée par un ensemble de solveurs numériques. Les méthodologies pour la discrétisation et résolution dépendent étroitement du régime ciblé.

### **4.2 Méthode de résolution d'équations pour la simulation thermique en régime transitoire**

Dans le cas d'un régime transitoire, une discrétisation spatiale sur les trois dimensions à l'aide de la méthode des différences finies est effectuée. L'équation des échanges de chaleur résultante est de la forme :

$$C\dot{T} + GT = P \quad (4.1)$$

Où  $G$  est la matrice de conductance,  $C$  la matrice de capacitance et  $P$  le vecteur de puissance. La matrice  $G$  décrit les échanges thermiques entre chaque cellule du maillage et ses cellules voisines, c'est une matrice en bandes inversible. La matrice  $C$  est une matrice diagonale. Le vecteur  $P$  contient les valeurs de puissance générée par chaque cellule du maillage.  $T$  est le vecteur de température.

L'équation 4.1 est une équation différentielle linéaire du premier ordre par rapport à la variable de temps. Une discrétisation temporelle basée sur la méthode d'Euler arrière permet de ramener cette

équation à une équation linéaire. En divisant le temps continu en intervalles discrets et en appliquant la méthode d'Euler arrière, on obtient, à chaque pas de temps  $\Delta t$ , l'équation suivante :

$$(\mathbf{I} + \Delta t \mathbf{C}^{-1} \mathbf{G}) \mathbf{T}_{m+1} = \mathbf{T}_m + \Delta t \mathbf{C}^{-1} \mathbf{P}_m \quad (4.2)$$

Où  $\mathbf{I}$  est la matrice identité,  $\mathbf{C}^{-1}$  l'inverse de la matrice  $\mathbf{C}$ ,  $\mathbf{G}$  la matrice de conductance.  $\mathbf{P}_m$  et  $\mathbf{T}_m$  sont respectivement le vecteur puissance et température au  $m$ -ième pas de temps. Le vecteur  $\mathbf{T}_{m+1}$  est le vecteur température au pas de simulation  $m+1$ .

À l'instant  $t_{m+1}$ , la température lors du pas temporel précédent  $t_m$  est déjà connue, le terme  $\mathbf{T}_m + \Delta t \mathbf{C}^{-1} \mathbf{P}_m$  est alors connu. Cette discrétisation temporelle permet d'avoir ainsi un système matriciel sous la forme  $\mathbf{Ax}=\mathbf{b}$ , avec  $\mathbf{A} = (\mathbf{I} + \Delta t \mathbf{C}^{-1} \mathbf{G})$ ,  $\mathbf{b} = \mathbf{T}_m + \Delta t \mathbf{C}^{-1} \mathbf{P}_m$  et  $\mathbf{x} = \mathbf{T}_{m+1}$ . Mais inverser le facteur  $(\mathbf{I} + \Delta t \mathbf{C}^{-1} \mathbf{G})$  est très coûteux pour les systèmes à grandes tailles. L'outil ICTherm utilise une méthode qui combine une décomposition des opérateurs pour résoudre efficacement l'équation 4.2 suivie d'une correction de l'erreur de décomposition pour augmenter la précision des résultats.

#### 4.2.1 Décomposition des opérateurs

Le principe de cette méthode est de ramener le système d'équations tridimensionnelles à trois systèmes unidimensionnels. Pour ce faire, la matrice de conductance  $\mathbf{G}$  doit être décomposée en trois matrices suivant chaque axe de projection  $x$ ,  $y$  et  $z$ . L'équation 4.2 est alors écrite sous la forme suivante :

$$(\mathbf{I} + \Delta t \mathbf{C}^{-1} \mathbf{G}_x) \mathbf{T}_{m+\frac{1}{3}} = \mathbf{T}_m + \Delta t \mathbf{C}^{-1} \mathbf{P}_m \quad (4.3)$$

$$(\mathbf{I} + \Delta t \mathbf{C}^{-1} \mathbf{G}_y) \mathbf{T}_{m+\frac{2}{3}} = \mathbf{T}_{m+\frac{1}{3}} \quad (4.4)$$

$$(\mathbf{I} + \Delta t \mathbf{C}^{-1} \mathbf{G}_z) \mathbf{T}_{m+1} = \mathbf{T}_{m+\frac{2}{3}} \quad (4.5)$$

Les nouvelles matrices  $\mathbf{G}_x$ ,  $\mathbf{G}_y$  et  $\mathbf{G}_z$  sont tridiagonales, chaque nouveau système d'équations est alors résolu avec l'algorithme de Thomas [24]. Les termes  $\mathbf{T}_{m+\frac{1}{3}}$  et  $\mathbf{T}_{m+\frac{2}{3}}$  sont les températures intermédiaires nécessaires au calcul de la température au pas du temps  $m+1$ .

### Algorithme de Thomas

L'algorithme de résolution d'une matrice tridiagonale ou algorithme de Thomas est une forme simplifiée d'élimination gaussienne pour les systèmes d'équations tridiagonaux. Il est basé sur une décomposition LU d'une matrice tridiagonale. Il a une complexité en  $O(n)$  où  $n$  est la taille du système à résoudre. Un pseudo-code de l'algorithme est donné dans l'Algorithme 4.1.

---

#### Algorithme 1 : Algorithme de Thomas

---

**Input** :  $A \in \mathbb{R}^{n \times n}$  telle que  $a_{i,j} \neq 0$  si  $|i - j| \leq 1$ ,  $b \in \mathbb{R}^n$   
**Output** :  $x \in \mathbb{R}^n$  tel que  $Ax = b$

```

1  $a_{0,0} \leftarrow 1/a_{0,0}$  ;
2 for  $i = 2 \dots n$  do % Factorisation LU
3    $a_{i,i-1} \leftarrow a_{i,i-1} * a_{i-1,i-1}$  ;
4    $a_{i,i} \leftarrow 1/(a_{i,i} - a_{i,i} * a_{i-1,i})$  ;
5 end
6 for  $i = 2 \dots n$  do % Substitution avant
7    $x_i \leftarrow x_i - a_{i,i} * x_{i-1}$  ;
8 end
9  $x_n \leftarrow x_n * a_{n,n}$  ;
10 for  $i = n - 1 \dots 1$  do % Substitution arrière
11    $x_i \leftarrow (x_i - a_{i,i+1} * x_{i+1}) * a_{i,i}$  ;
12 end
```

---

Algorithme 4.1 : Pseudo-code de l'algorithme de Thomas [4]

### 4.2.2 Schéma itératif de correction de l'erreur de décomposition

L'étape de décomposition des opérateurs permet d'assurer efficacement la simulation thermique des circuits intégrés tridimensionnels pour le régime transitoire, mais cette décomposition introduit une petite erreur qui sera corrigée par un schéma itératif de correction basé sur le principe des itérations de Richardson [24].

### 4.2.3 Algorithme complet de la méthode proposée

Un pseudo-code de la méthode proposée pour la simulation du régime transitoire est donné par l'Algorithme 4.2. À chaque pas de temps, l'algorithme de Thomas est utilisé pour résoudre le système d'équations obtenus; le vecteur température obtenu est utilisé comme température de départ pour le schéma itératif de correction de l'erreur de décomposition afin d'augmenter la précision des résultats obtenus.



---

**Algorithme 2** : Méthode proposée pour la simulation du régime transitoire

---

**Input** :  $C, G, (P_m)_{m \in \mathbb{N}}, X_0$  : température initiale,  $\Delta t, \epsilon$  : précision,  $s$  : nombre de pas à simuler

**Output** :  $(X_m)_{m \in \mathbb{N}}$  : vecteurs température aux instants  $t_m = m\Delta t$

```

1 for  $m = 1 \dots s$  do /* Pour chaque pas de temps */
2    $X_m \leftarrow X_{m-1} + \Delta t C^{-1} P_{m-1}$ ;
3    $X_m \leftarrow (I + \Delta t C^{-1} G_x)^{-1} X_m$ ;
4    $X_m \leftarrow (I + \Delta t C^{-1} G_y)^{-1} X_m$ ;
5    $X_m \leftarrow (I + \Delta t C^{-1} G_z)^{-1} X_m$ ;
6    $\tilde{X}_m \leftarrow 0$ ;
7   while  $\|X_m - \tilde{X}_m\| > \epsilon$  do /* Schéma itératif pour corriger l'erreur de
   décomposition */
8      $\tilde{X}_m \leftarrow X_m$ ;
9      $X_m \leftarrow -\Delta t C^{-1} (G_y + G_z) X_m$ ;
10     $X_m \leftarrow X_m + X_{m-1} + \Delta t C^{-1} P_{m-1}$ ;
11     $X_m \leftarrow (I + \Delta t C^{-1} G_z)^{-1} X_m$ ;
12  end
13 end

```

---

Algorithme 4.2: Pseudo-code de l'algorithme de résolution du régime transitoire proposé [4]

### 4.3 Méthode de résolution d'équations pour la simulation thermique en régime permanent

Dans le régime permanent, une discrétisation spatiale est suffisante pour obtenir un système sous la forme  $Ax=b$ , puisque le paramètre temps n'est plus considéré. Comme dans le cas du régime transitoire, la méthode des différences finies est retenue pour ses propriétés de stabilité. Après cette phase de discrétisation, l'équation de chaleur lors d'un régime permanent s'écrit sous la forme :

$$GT = P \quad (4.6)$$

Où  $G$  est la matrice de conductance, c'est une matrice en bandes<sup>1</sup> qui décrit les échanges thermiques entre chaque cellule du maillage et ses cellules voisines.  $P$  est le vecteur puissance.  $T$  représente la température de chaque cellule du maillage quand le régime stationnaire est atteint. Deux algorithmes distincts sont proposés pour résoudre efficacement ce système d'équations, pour les systèmes de composition solide pure et ceux qui intègrent des géométries fluidiques.

---

<sup>1</sup> Matrice creuse dont les coefficients non nuls se regroupent autour de la diagonale

### 4.3.1 Cas d'un circuit solide

Lorsque le circuit intégré tridimensionnel est constitué de composants et sous composants faits de matériaux solides uniquement, les échanges de chaleur sont symétriques, la matrice de conductance  $G$  résultante de la discrétisation de ce système est alors symétrique. Dans ce cas de figure, la méthode de projection numérique, gradient conjugué (GC) est adoptée par l'outil ICtherm.

#### Méthode du gradient conjugué

La méthode du gradient conjugué [24] est une méthode numérique itérative pour résoudre les systèmes d'équations linéaires où la matrice est symétrique définie positive. Elle a été introduite dans les années cinquante par Cornelius Lanczos et Magnus Hestenes. Combinée à un pré-conditionnement du système linéaire, elle constitue la méthode la plus performante connue pour la résolution de systèmes matriciels de grande taille.

Algorithme :

---

<b>Algorithme 3 : Méthode du gradient conjugué</b>	
<b>Input :</b> $A \in \mathbb{R}^{n \times n}$ , $b \in \mathbb{R}^n$ , $\epsilon$ (précision)	
<b>Output :</b> $x \in \mathbb{R}^n$ tel que $\ b - Ax\  < \epsilon$	
1	<b>Initialisation :</b> $x \leftarrow 0$ ; $r \leftarrow b$ ; $p \leftarrow r$ ;
2	<b>while</b> $\ r\  > \epsilon$ <b>do</b>
3	$q \leftarrow Ap$ ;
4	$\alpha \leftarrow \frac{\langle r r \rangle}{\langle q p \rangle}$ ;
5	$\beta \leftarrow \frac{1}{\ r\ }$ ;
6	$x \leftarrow x + \alpha p$ ;
7	$r \leftarrow r - \alpha q$ ;
8	$\beta \leftarrow \beta * \ r\ $ ;
9	$p \leftarrow r + \beta p$ ;
10	<b>end</b>

---

Algorithme 4.3: Pseudo-code de l'algorithme du gradient conjugué [4]

Dans l'Algorithme 4.3, un pseudo-code de la méthode du gradient conjugué est donné. Les calculs sont réitérés jusqu'à la convergence.

Les principaux avantages de l'algorithme sont :

- Permet de résoudre les matrices de grandes tailles
- Stable : convergence assurée en au plus  $n$  itérations ;  $n$  étant la taille du système matriciel
- Complexité : linéaire

### 4.3.2 Cas d'un circuit contenant des composants fluidiques

Lorsque le circuit intégré tridimensionnel est refroidi par écoulement d'un fluide, le transport de chaleur par le fluide se fait dans le sens de son écoulement; la matrice de conductance  $G$  résultante de la discrétisation de ce système est alors asymétrique. Dans ce cas, l'outil ICTherm utilise la méthode itérative : généralisation de la méthode de minimisation du résidu (ou GMRES) [24], pour résoudre efficacement l'équation de chaleur.

#### Méthode du GMRES

La méthode de généralisation de la méthode de minimisation du résidu (ou GMRES) est développée par Yousef Saad et Martin H. Schultz en 1986. C'est une méthode numérique de projection qui donne une approximation de la solution par un vecteur appartenant à un espace de Krylov avec un résidu minimal.

Algorithme :

---

**Algorithme 4 : Méthode GMRES**

---

**Input :**  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$ ,  $\epsilon$  (précision),  $m$  (redémarrage)

**Output :**  $x \in \mathbb{R}^n$  tel que  $\|b - Ax\| < \epsilon$

```

1 Initialisation :  $x \leftarrow 0$ ;  $r \leftarrow b$ ;
2 while  $\|r\| > \epsilon$  do
3    $v_1 \leftarrow r/\|r\|$ ;
4   for  $j=1,\dots,m$  do
5      $w \leftarrow Av_j$ ;
6     for  $i=1,\dots,j$  do
7        $h_{i,j} \leftarrow w^T v_i$ ;
8        $w \leftarrow w - h_{i,j}v_i$ ;
9     end
10     $h_{j+1,j} \leftarrow \|w\|_2$ ;
11     $v_{j+1} \leftarrow w/\|w\|_2$ ;
12  end
13   $V_m \leftarrow [v_1 \ v_2 \ \dots \ v_m]$ ;
14   $x \leftarrow x + V_m y_m$  où  $y_m$  minimise  $\|r - AV_k y\|_2$ ;
15   $r \leftarrow b - Ax$ ;
16 end

```

---

Algorithme 4.4: Pseudo-code de l'algorithme GMRES [4]

Un pseudo-code de GMRES est proposé dans l’Algorithme 4.4. La rapidité de la convergence de l’algorithme dépend de la valeur de précision  $\varepsilon$  et du facteur de redémarrage  $m$ .

Les principaux avantages de l’algorithme sont :

- Permet de résoudre les matrices de grandes tailles
- Stable : convergence lente résolue par un préconditionnement.
- Complexité :  $O(nm)$  :  $n$  étant la taille du système matriciel et  $m$  le facteur de redémarrage.

### 4.3.3 Préconditionnement des algorithmes

Pour accélérer la convergence des algorithmes de gradient conjugué et GMRES, une étape de préconditionnement de la matrice de conductance  $G$  est préconisée. Une décomposition LU incomplète sera donc appliquée sur la matrice avant de procéder à la résolution du système. Un pseudo-code de la méthode de décomposition incomplète ILU est donné par l’Algorithme 4.5.

---

**Algorithme 5 : Algorithme ILU(0)**

---

Input :  $A \in \mathbb{R}^{n \times n}$   
Output :  $L, U \in \mathbb{R}^{n \times n}$  (stockées dans A)

```

1 for  $k=1, \dots, n-1$  do
2   for  $i=k+1, \dots, n$  do
3     if  $(i, k) \in NZ$  then
4        $a_{ik} \leftarrow a_{ik} / a_{kk}$  ;
5       for  $j=k+1, \dots, n$  do
6         if  $(i, j) \in NZ$  then
7            $a_{ij} \leftarrow a_{ij} - a_{ik} * a_{kj}$  ;
8         end
9       end
10    end
11  end
12 end

```

---

Algorithme 4.5: Pseudo-code de l’algorithme de décomposition incomplète ILU [4]

## 4.4 Synthèse

Cette partie présente les résultats obtenus lors de la validation des méthodologies énoncées dans les sections précédentes, trace un bilan des solveurs proposés et souligne les limitations du travail précédemment réalisé sur ICTherm.

#### 4.4.1 Validation et résultats

La méthodologie utilisée est validée en comparant les résultats obtenus par simulation à des mesures thermiques effectuées sur un circuit physique réel et modélisation avec l'outil Comsol. Un système composé d'une carte mère contenant un processeur Intel Atom, d'une cellule de refroidissement de Peltier et d'une caméra thermique permet de relever la température de surface du processeur Atom durant son fonctionnement par imagerie infrarouge.

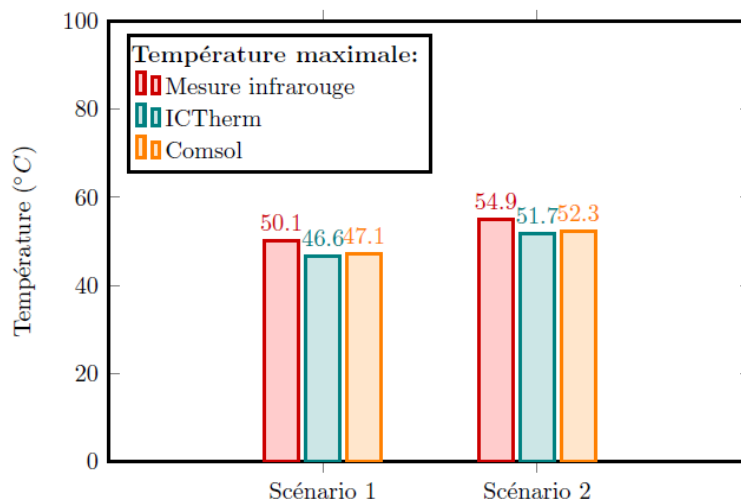


Figure 4.1: Comparaison entre les températures obtenues par imagerie et par simulation [4]

Les résultats obtenus par simulation avec ICTherm et Comsol et par mesure sont illustrés dans la Figure 4.1. Deux scénarios ont été considérés, ils correspondent respectivement à une activité du processeur considéré à 90% et 100%. Pour ces scénarios, la différence entre les simulations faites par ICTherm et Comsol est inférieure à 1%, la différence entre valeur de la température maximale mesurée et simulée est inférieure à 6% [4].

#### 4.4.2 Résumé des méthodologies

Les méthodologies de discrétisation et de résolution suivies par ICTherm pour le régime permanent et transitoire sont résumées dans la Figure 4.2. Celle-ci reprend la représentation illustrée dans la Figure 3.1 et positionne les méthodes retenues (marquées en rouge) par rapport à la littérature. ICTherm se base sur la méthode des différences finies pour la discrétisation spatiale et la méthode d'Euler arrière pour la discrétisation temporelle. Les solveurs GMRES et GC sont utilisés pour la

résolution du régime permanent tandis qu'une décomposition des opérateurs suivie d'un schéma itératif de correction d'erreur est retenue pour le régime transitoire.

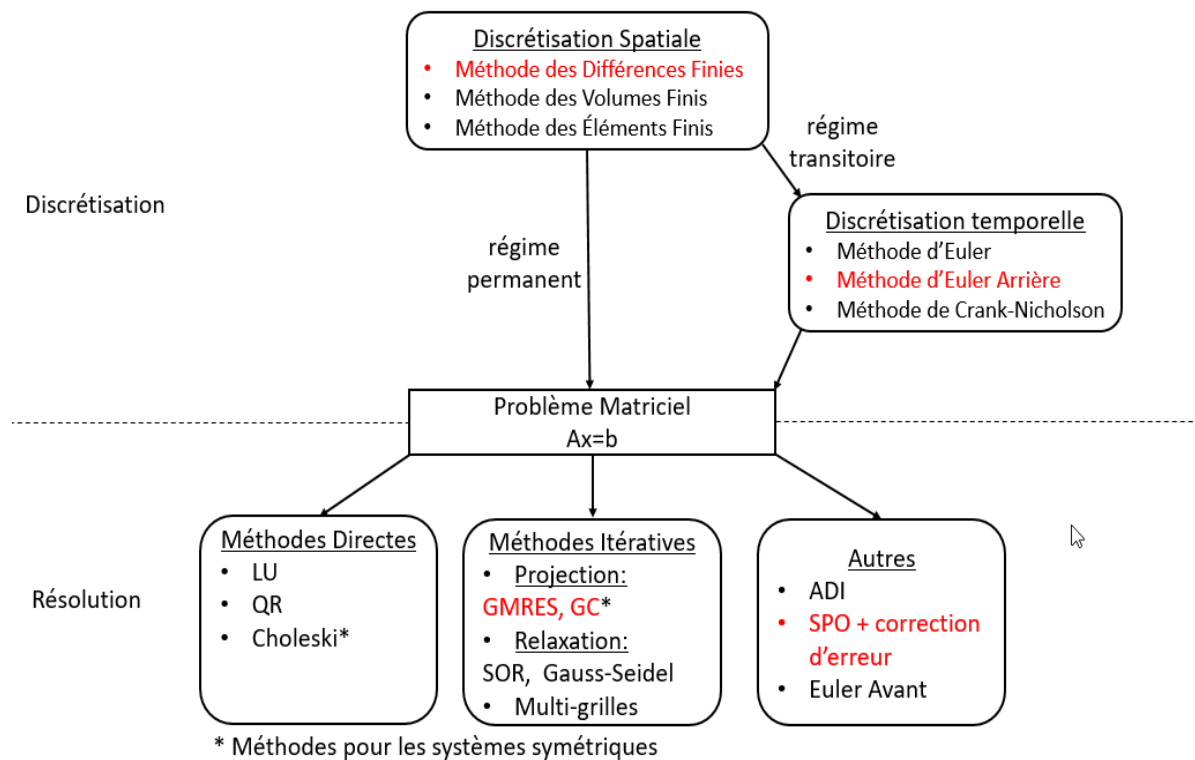


Figure 4.2: Méthodologies sélectionnées par rapport à l'état de l'art

La convergence de l'algorithme du gradient conjugué est assurée en au plus  $n$  itérations ( $n$  étant la taille du système matriciel) mais dans la pratique grâce au preconditionnement LU la convergence est assez rapide. Lorsqu'on utilise l'algorithme du GMRES, il est redémarré toutes les 100 itérations. L'expérience montre que ce paramètre de redémarrage offre un bon compromis entre la vitesse de convergence et la complexité de la méthode [4].

#### 4.4.3 Limitations

Des implémentations des méthodologies ont été réalisées par Alain Fourmigue pour valider les choix de solveurs. Toutefois ces implémentations présentent plusieurs limitations : dans certains cas, il s'agit de code en dur pour modéliser un circuit en particulier ; dans d'autres, un code peu flexible pour simuler un type particulier de circuit pour le régime permanent. Pour modéliser un

plus grand nombre de types de circuit et surtout les circuits futurs, il faudra généraliser et adapter les méthodologies proposées et revoir la modélisation des circuits.

## CHAPITRE 5 ICTHERM : DÉFINITION ET MISE EN PLACE DE L'OUTIL

Ce chapitre présente le travail effectué dans le cadre de nos travaux de recherche. Les premières sections introduisent l'outil ICTherm, ses caractéristiques et son architecture. Par la suite, une présentation détaillée des modules : entrée, modélisation, maillage, discrétisation, résolution et sortie; est donnée.

### 5.1 L'outil ICTherm

La Figure 5.1 illustre le flot de l'outil ICTherm. Le simulateur ICTherm prend en entrée une description géométrique du circuit intégré à simuler, des matériaux le composant, des sources de chaleur, des puissances générées et des paramètres de simulation. Il génère en sortie la carte thermique résultante.

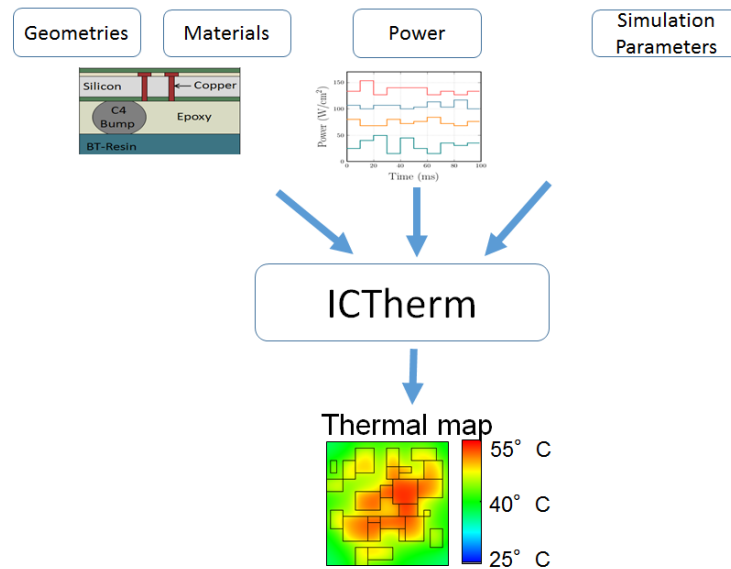


Figure 5.1: Flot de l'outil ICTherm

### 5.2 Caractéristiques d'un outil de simulation efficace

Le but de notre travail est d'implémenter un outil générique de simulation thermique des circuits intégrés tridimensionnels à partir des travaux théoriques décrits dans le chapitre précédent. Le simulateur sera un outil gratuit à code source ouvert (ou en anglais open source) visant ainsi un déploiement pour la communauté académique et pour l'industrie.



Notre apport sera d'assurer pour l'outil développé les caractéristiques suivantes :

- Sa genericité : l'outil doit permettre la simulation de tout type de circuit intégré tridimensionnel existant pour le régime transitoire et permanent.
- Son aspect évolutif : l'outil doit permettre de prendre en considération les nouvelles fonctionnalités offertes par les nouveaux circuits intégrés pour pouvoir les modéliser et simuler efficacement et pouvoir suivre leur évolution.
- Sa facilité d'utilisation : l'outil doit être ergonomique, les entrées facilement définies et les sorties facilement interprétables et utilisables.
- Son aspect personnalisable : notre collaboration avec l'université de Sherbrooke sur un futur circuit intégré avec des composants et paramètres spécifiques et notre exploration d'un circuit optique nous a permis de constater que certains concepteurs ont des besoins spécifiques. L'idée de leur permettre d'intégrer quelques modules propres à leur circuit les aidera à mieux faire avancer leur conception.

Dans chaque partie de description de notre travail, on mettra l'accent sur la justification de ces critères.

## **5.3 Défis des circuits futurs**

Au cours de notre travail de recherche, nous avons travaillé, dans le cadre de collaborations avec l'université de Sherbrooke et l'école centrale de Lyon, sur deux circuits de nouvelles générations en cours de conception.

### **5.3.1 Circuit à refroidissement microfluidique du projet STREAMS**

Dans le cadre du projet STREAMS (Smart Technologies for energy Efficient Active cooling in advanced Microelectronic Systems), l'équipe du professeur Luc Fréchette à l'université de Sherbrooke participe à la conception d'une nouvelle génération de circuits intégrés tridimensionnels à refroidissement multifluidique[33]. Il s'agit d'un circuit doté de microvalves activées thermiquement ce qui permet de contrôler le débit des cellules pour une meilleure régulation de la température. Le débit du fluide utilisé, qui pour des circuits classiques est un paramètre fixe de la simulation, est un paramètre variable à explorer dans ce cas de figure, ce qui représente un défi pour la modélisation et la simulation. Un second défi de ce circuit est la

géométrie non uniforme des microcanaux prévus pour faire circuler le liquide de refroidissement (l'eau dans notre cas). La Figure 5.2 représente le circuit multifluidique et les microvalves thermiquement activées.

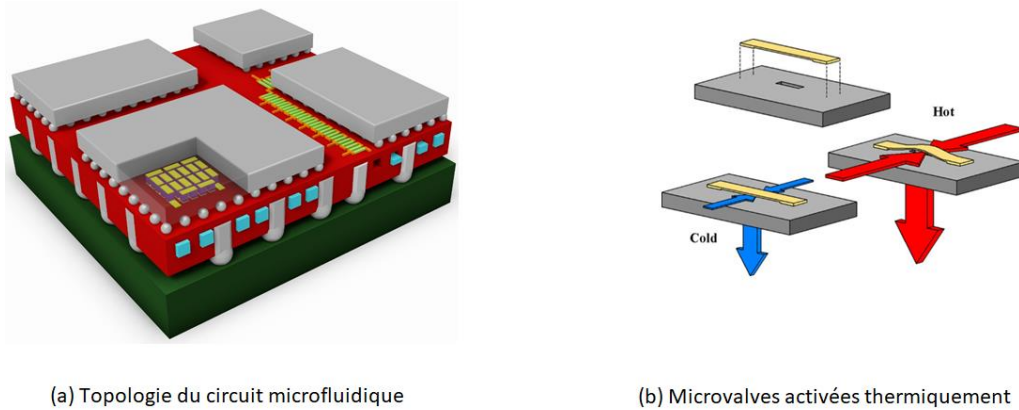


Figure 5.2 Circuit à refroidissement microfluidique du projet STREAMS [33]

### 5.3.2 Circuit intégrant des composants optiques

Dans le cadre d'une collaboration avec l'équipe du professeur Sébastien Le Beux de l'école centrale de Lyon, nous avons travaillé sur la modélisation et la simulation de leur circuit en cours de conception. Il s'agit d'un circuit électrique qui intègre des composants optiques illustré dans la Figure 5.3.

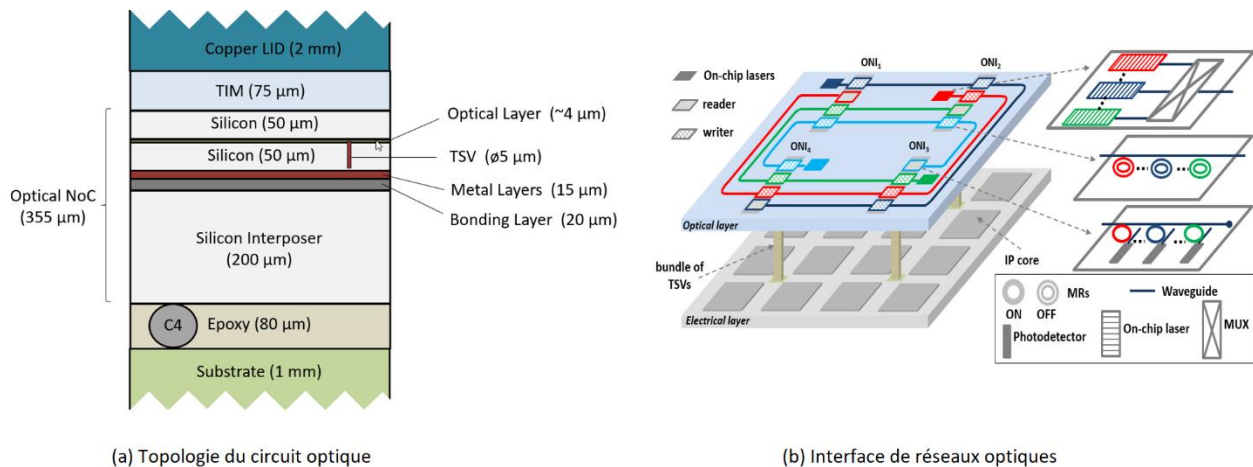


Figure 5.3 Circuit intégré tridimensionnel intégrant des composants optiques [34]

Une simulation thermique précise est très importante pour ce type de circuit puisque les composants optiques sont très sensibles à la chaleur. La partie optique est formée par des interfaces de réseaux optiques ONI (venant de l'anglais Optical Network Interface) formées par des matrices de microlasers. Pour ce circuit, une simulation efficace et précise permettra d'explorer l'effet de la distance entre les lasers et les puissances attribuées sur le système optique.

## **5.4 Définitions des besoins et flot de définition du simulateur ICTherm**

Pour définir l'outil ICtherm, notre point de départ était les solveurs numériques détaillés dans le Chapitre 4. Ce travail précédemment réalisé n'est pas exploitable à l'état actuel, un travail de recherche sur la modélisation des circuits en entrée et d'implémentation a été réalisé :

- Pour simuler des circuits de tout type (solide, microfluidique, optique) pour les régimes permanent et transitoire, nous devons pouvoir facilement détecter chaque cas et sélectionner les solveurs adéquats.
- Pour pouvoir représenter des circuits hétérogènes, avec des géométries complexes, faits de multiples composants de dimensions variables et avec des caractéristiques thermiques diverses, une phase de modélisation fine et efficace est réalisée.
- Pour assurer une étape de discrétisation efficace de l'équation de chaleur, une définition d'un maillage adéquat et dépendant de la géométrie du circuit est faite.
- Pour mettre en place l'outil ICTherm au complet, les travaux (de maillage, de modélisation ...) réalisés ont été implémentés et enrichis d'un interfaçage offrant plus de flexibilité et simplicité.

Une fois l'outil ICTherm mis en place, nous avons travaillé sur le raffinement et l'amélioration de chacune des phases définies :

- Pour permettre la modélisation et simulation de circuits de nouvelles générations, les phases de maillage et modélisation que nous avons définies ont été améliorées; nous avons enrichi les solveurs utilisés à travers un schéma itératif de la phase de discrétisation et résolution.
- Pour avoir des résultats facilement interprétables, l'outil est enrichi d'un module de sortie.

La Figure 5.4 illustre ces phases de définition, de modélisation et de mise en place de l’outil ICTherm régies par les besoins d’une simulation générique, complète, simple et efficace.

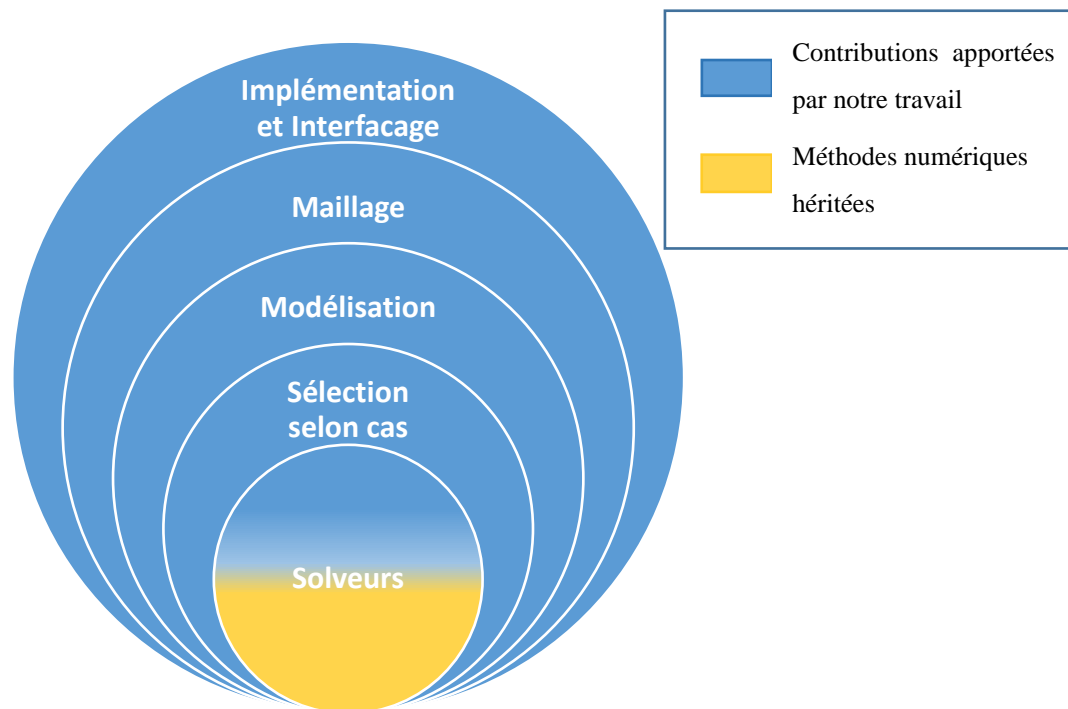


Figure 5.4 Flot de définition et modélisation du simulateur ICTherm

## 5.5 Le choix du langage de développement

Nous avons développé le simulateur ICTherm à l’aide du langage Python.

### 5.5.1 Python

Python [35] est un langage de programmation qui supporte à la fois la programmation orientée objet, la programmation procédurale et la programmation fonctionnelle. Il a été créé par le développeur néerlandais Guido Van Rossum qui sortit la première version de Python en 1991. Python est un langage interprété : il ne doit donc pas être compilé en un langage machine avant d’être exécuté. Les programmes sont exécutés directement par l’interpréteur Python.

Python est supporté par une large communauté de contributeurs qui participent à améliorer le langage et à créer de bibliothèques tierces qui s’étendent des frameworks web aux systèmes embarqués. Le langage Python est un langage dont la syntaxe est à la fois simple à adopter et riche en termes

de fonctionnalités. Sa simplicité lui a valu d'être largement adopté dans plusieurs secteurs, notamment dans la recherche scientifique.

Les projets à code source ouvert développés en Python proposent des bibliothèques riches et complètes qui couvrent des sujets scientifiques comme :

- Les calculs scientifiques : SciPy intègre des modules d'algèbre linéaire, d'optimisation, d'intégration ainsi que des fonctions de traitement de signal et d'images.
- Le machine learning : un ensemble de modules pour l'analyse de données et le data mining.
- L'affichage de courbes et graphes statistiques.

Python bénéficie aussi du fait qu'il soit multiparadigme et de ce fait permet une grande flexibilité pour écrire des programmes simples ou beaucoup plus complexes. On peut l'adopter pour écrire des programmes procéduraux simples, mais tout aussi bien des architectures orientées objet complexes.

En contrepartie, le fait que Python soit un langage interprété réduit de peu ses performances d'exécution comparées aux langages compilés. Cependant, plusieurs bibliothèques et modules sont optimisés pour fonctionner à des vitesses relativement proches des langages compilés comme le langage C.

### 5.5.2 Le choix du Python

Python est un langage à la fois facile à utiliser et très riche puisque plusieurs bibliothèques sont déjà fournies, ce qui va permettre d'accélérer le développement. La plupart des algorithmes présentés au Chapitre 4 sont déjà existants dans la bibliothèque d'algèbre linéaire de Python comme l'algorithme GMRES et gradient conjugué. De plus plusieurs structures de données sont déjà fournies pour le stockage et la manipulation des matrices creuses ; la matrice de conductance  $G$  est une matrice creuse, permettant ainsi une optimisation dans la gestion de la mémoire.

La facilité du codage en Python et son aspect orienté objet favorise une bonne structuration du code permettant facilement sa mise à jour et l'ajout de nouvelles fonctionnalités et modules. Cet aspect est très important pour le critère évolutif de l'outil, en effet notre travail vise à pouvoir facilement ajouter des modules pour répondre à l'évolution rapide des circuits intégrés tridimensionnels. De plus, les outils de simulations thermiques sont généralement utilisés par les concepteurs qui ont peu d'expérience avec les langages de programmation tels que C ou C++, mais plus avec du Matlab,

et pour leur laisser la possibilité d'ajouter quelques fonctionnalités à ICTherm il est primordial d'adopter un langage simple à apprendre et comprendre ; ce qui est primordial pour l'aspect personnalisable de l'outil.

## 5.6 Architecture de l'outil

La Figure 5.5 présente l'architecture de l'outil ICTherm et ses différents modules : entrée, modélisation, maillage, discrétisation, résolution et sortie.

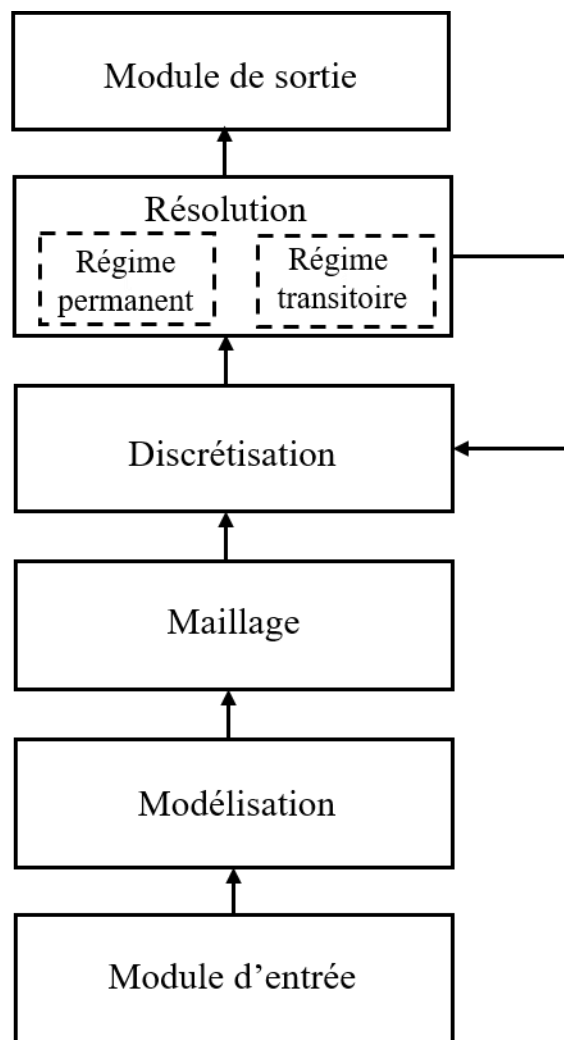


Figure 5.5: Architecture de l'outil ICTherm

Les modules discrétisation et résolution sont basés sur les solveurs numériques présentés précédemment dans le Chapitre 4, les premières versions d'ICTherm ont donné un code en dur pas flexible et non générique. Notre contribution sur cette partie est d'offrir une réimplémentation uniforme et générique en Python du simulateur ICTherm et de généraliser la méthodologie utilisée à tout type de circuit.

Des modules : entrée, modélisation, maillage et sortie ont été ajoutés pour enrichir l'outil ICTherm en lui apportant de la flexibilité, de la généricité et de la facilité d'utilisation.

Tous les modules présentés seront détaillés dans les sections suivantes.

## 5.7 Le module d'entrée

Le module d'entrée de l'outil est l'interfaçage avec l'utilisateur. Dans le cas de notre outil, il est primordial pour généraliser et permettre la modélisation de tout type de circuit. Dans une première version, les données sont saisies à l'aide d'un fichier de configuration normalisé. Dans un souci de rendre la tâche de décrire son circuit et ses paramètres plus facile pour le concepteur, une interface graphique a enrichi ce module d'entrée. Cette interface a été développée en collaboration avec un stagiaire. Bien qu'un outil soit performant, sa complexité peut entraver son déploiement. Une interface ergonomique et simple à utiliser renforcera alors l'utilisation d'ICTherm. Cette interface assurera le lien entre l'utilisateur et le module de modélisation et offrira à l'utilisateur l'outillage nécessaire pour décrire son circuit et sa composition d'une meilleure façon. Pour accentuer la flexibilité et l'évolutivité de l'outil, nous avons opté pour un modèle MVC (Modèle, Vue et Contrôleur) pour pouvoir mettre à jour, valider, changer et déboguer les modules de l'outil séparément. L'interface est la représentation graphique du modèle utilisé pour décrire le système à résoudre. Celle-ci reflète donc les éléments de modélisation du circuit (bloc, matériau et paramètres de simulation).

Les figures : Figure 5.6, Figure 5.7, Figure 5.8 montrent les fenêtres graphiques qui servent à manipuler les entités blocs, matériaux et la configuration de la simulation pour une modélisation rapide et efficace du circuit intégré à simuler.

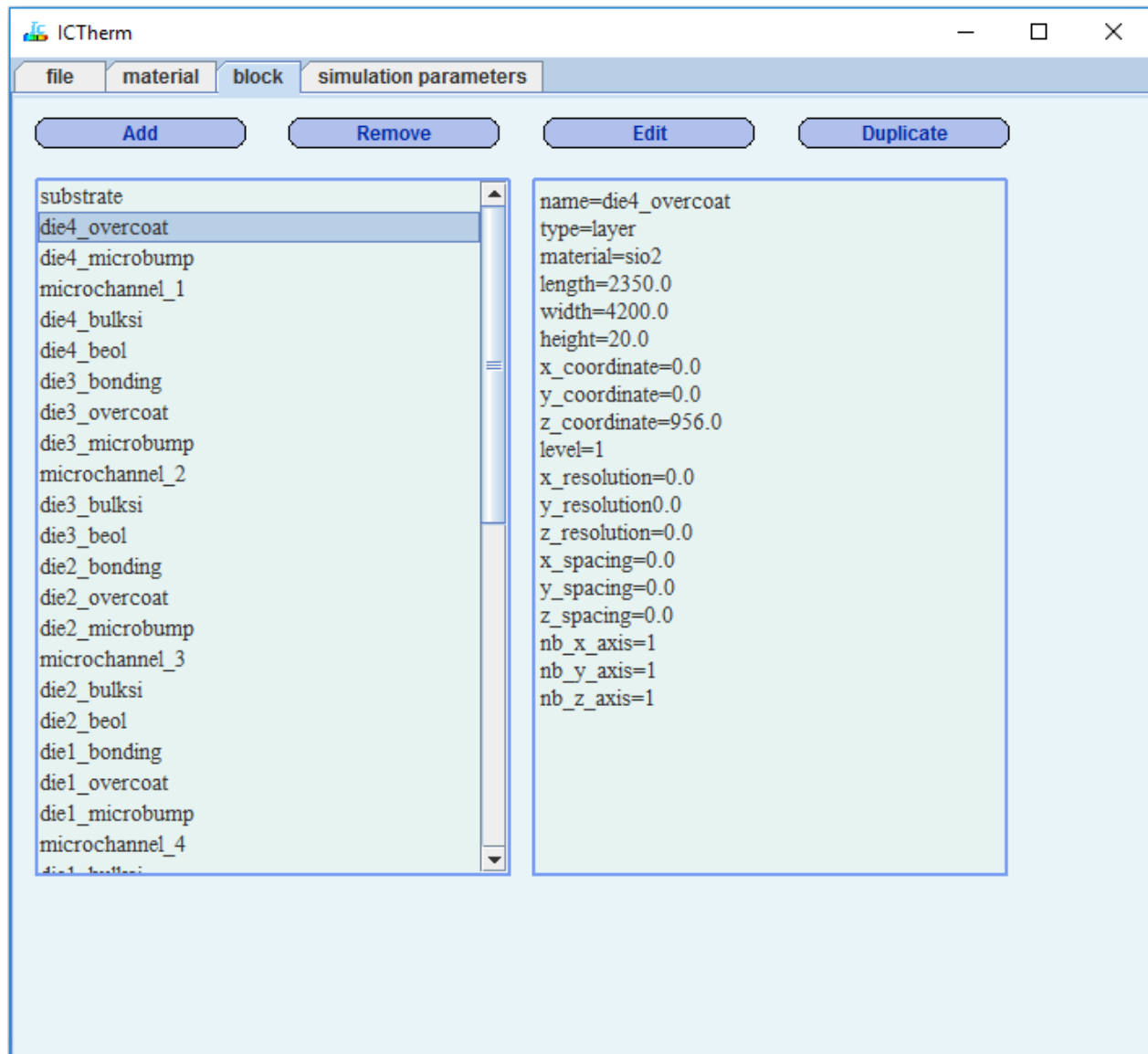


Figure 5.6: Fenêtre pour ajouter, dupliquer et mettre à jour les géométries du circuit



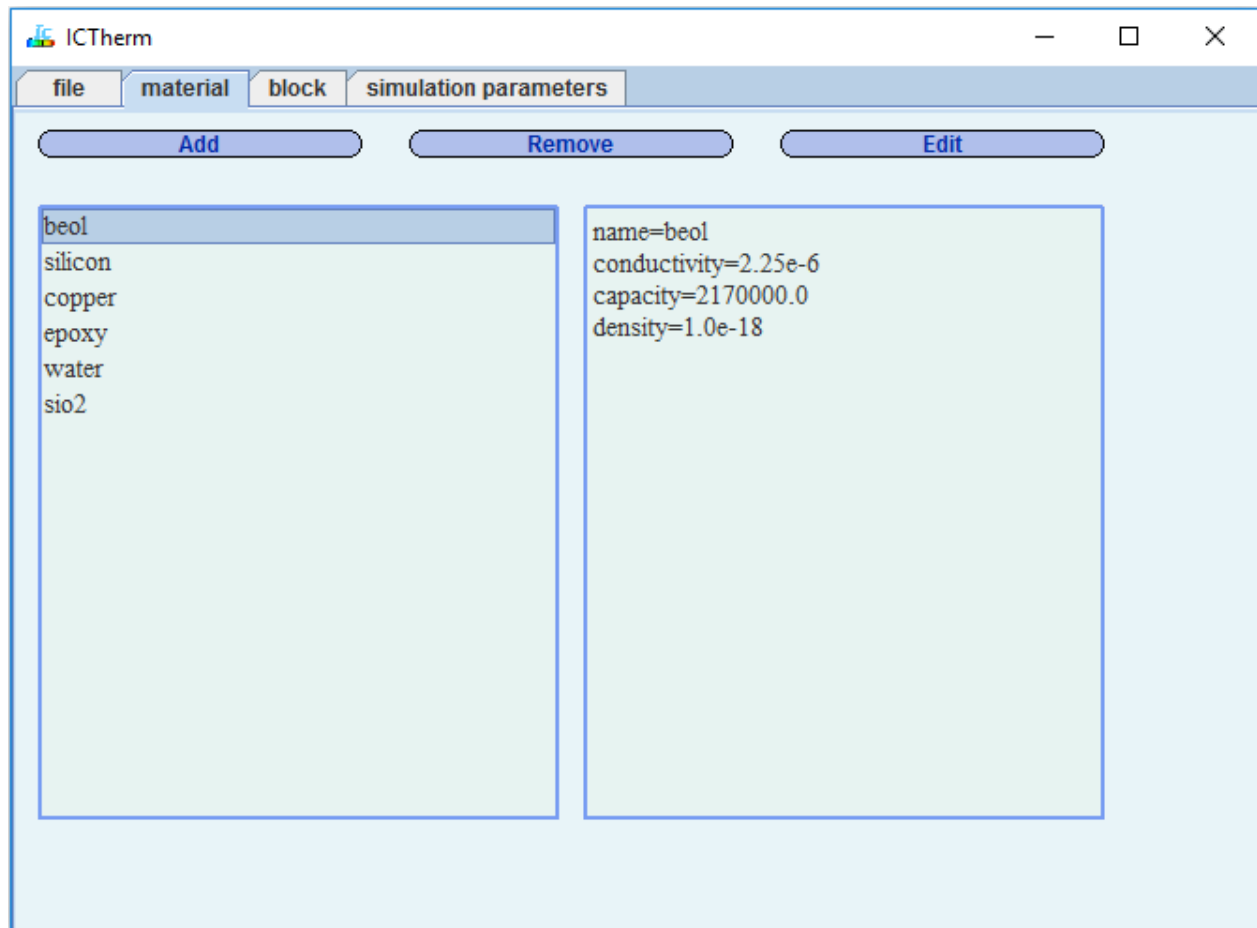


Figure 5.7: Fenêtre pour configurer les matériaux dont les composants du circuit sont faits

Parameter	Value
initial temperature	300.0
state *	transient
h_air_top	0.0
h_air_bottom	0.0
h_air_north	0.0
h_air_south	0.0
h_air_east	0.0
h_air_west	0.0
mesh sizing	1
x resolution	0.0
y resolution	0.0
z resolution	0.0
time step *	0.0
duration *	0.0

Buttons: Validate, Clear

Figure 5.8: Fenêtre pour configurer les paramètres de simulation

## 5.8 La modélisation

Pour pouvoir simuler tout type de circuit intégré tridimensionnel, il faut que l'outil ICTherm puisse tout d'abord modéliser tous les détails dans le circuit. Cette modélisation est assurée par la partie frontale de l'outil. Pour assurer la genericité de l'outil, une formalisation de la description des données relatives à la géométrie du circuit, les puissances générées et les détails de simulation a été faite. L'adoption d'un format normalisé, complet et facilement extensible ajoute de la flexibilité et l'évolutivité au simulateur. Un fichier de type ini a été utilisé comme format intermédiaire pour la représentation des données sur le circuit.

Quatre informations principales sont indispensables pour une bonne modélisation des circuits intégrés :

- Les composants et les sous-composants le constituant ainsi que leurs dimensions.
- Les matériaux avec lesquels sont faits ces composants.
- Les puissances générées par les sources de chaleur.
- Les paramètres de simulation.

### **5.8.1 Modélisation des composants et géométries**

Chaque composant et sous-composant du circuit à simuler est modélisé par une entité de type bloc. Chaque bloc doit contenir un ensemble d'informations sur le composant qu'il modélise :

- Dimensions du composant : longueur, largeur et hauteur.
- Ses coordonnées : sur les trois plans dimensionnels.
- Le matériau dont il sera fabriqué.
- Son type : solide, microcanal ou source de chaleur.
- Des informations spécifiques aux types : le débit du fluide et le coefficient de transfert thermique pour les microcanaux, les valeurs de puissances pour les sources de chaleur.
- Des paramètres supplémentaires pour le support de quelques circuits particuliers.
- Des paramètres optionnels : définir le niveau de maillage du composant.

Cette modélisation permet de représenter toutes les géométries présentes dans le circuit.

La modélisation des sources de chaleur et la description énergétique de celles-ci se fait aussi à travers l'entité bloc, en effet une source de chaleur est vue comme un bloc qui génère une puissance non nulle. La puissance générée est représentée par un paramètre du bloc, qui peut prendre plus d'une valeur dans le cas d'un régime transitoire pour représenter une carte de puissance variable par pas de simulation.

### **5.8.2 Modélisation des matériaux**

Les caractéristiques des matériaux constituant les couches et les composants du circuit intégré à simuler définissent les transferts thermiques dans celui-ci. D'où l'importance de bien les modéliser. L'entité matériau offre une représentation complète de chaque matériau et de ses caractéristiques

thermiques. Cette entité encapsule les paramètres capacité, conductivité et densité du matériau. Le paramètre matériau des entités blocs doit correspondre à un matériau de l'entité.

### 5.8.3 Paramètres de simulation

Pour représenter le paramétrage de la simulation, le modèle est enrichi de quelques options telles que :

- Le régime de la simulation : transitoire ou permanent.
- Les paramètres du maillage qui seront détaillés dans la partie suivante.
- Les coefficients de transfert thermique entre le circuit et l'environnement où il évolue pour représenter les conditions aux limites.
- Le temps et les pas de simulation dans le cas d'un régime transitoire.

## 5.9 Le maillage

La discrétisation spatiale du circuit modélisé sur chacun des axes  $x$ ,  $y$  et  $z$  opère sur un maillage tridimensionnel. Pour créer ce maillage, deux contraintes étaient imposées dans ICTherm pour assurer l'applicabilité des méthodes choisies [4] : le maillage doit être orthogonal, structuré et conforme (ceci est dû à la méthode de discrétisation spatiale choisie) et chacune de ses cellules doit être faite dans un matériau unique. Le nombre de cellules du maillage définira la taille des matrices de conductance, capacitance, le vecteur puissance et le vecteur température. Pour les circuits intégrés tridimensionnels, ce nombre est de l'ordre de millions ( $10^6$ ) de cellules. Plus le nombre est important plus les résultats sont précis, mais plus les calculs sont lents et gourmands. Cette phase de maillage est donc très importante pour la suite des travaux, d'ailleurs trouver le maillage adéquat pour chaque problème est un sujet de recherches à part.

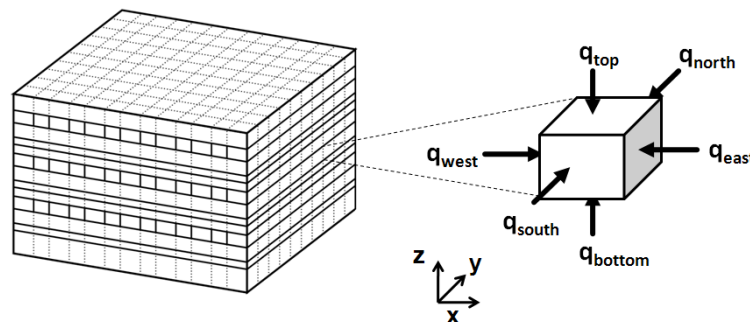


Figure 5.9: Un maillage orthogonal structuré conforme [36]

Dans notre travail, nous avons vu l'importance d'offrir divers niveaux de maillage pour prendre en compte les caractéristiques du circuit (circuit multifluidique, circuit optique) et les besoins de l'utilisateur (simulation grossière pour une première estimation ou simulation plus fine pour les sources de chaleur de taille réduite comme les lasers pour les circuits optiques).

Nous proposons les niveaux de maillage suivants :

- Un niveau basique ou grossier de maillage : ce niveau offre le maillage minimal qui respecte les contraintes d'orthogonalité et d'unicité du matériau par cellule.
- Un niveau de maillage régit par les dimensions des composants du circuit pour une précision des résultats plus élevée.
- Un niveau qui laisse à l'utilisateur le choix de fixer les dimensions maximales des cellules du maillage : ce niveau ajoute plus de flexibilité au concepteur de définir son propre maillage adapté à ses besoins.

Nous avons ajouté en combinaison avec ces niveaux, la possibilité de fixer un seuil de maillage par composants du circuit intégré. Cette option a été ajoutée pour permettre de mieux modéliser les circuits incluant des dimensions très différentes comme les circuits optiques qui intègrent des matrices de lasers de taille très réduite par rapport à la taille du circuit ; un maillage plus fin au niveau de ces lasers et plus grossier ailleurs permettait de mieux visualiser leur comportement thermique tout en gardant une taille du système raisonnable. Il est à noter que tous les niveaux définis offrent un maillage non uniforme ce qui permet de mieux modéliser tous les types de circuits et de faciliter la résolution du système d'équations obtenu.

## 5.10 La discrétisation

La discrétisation du système est réalisée par la création de la matrice de conductance et de capacitance. Une fois le maillage réalisé, les valeurs de conductance et capacitance sont calculées pour chaque cellule du maillage et sont régies par les types des échanges thermiques entre cette cellule et ses voisines afin d'atteindre l'équilibre thermique.

Soit la cellule *Cell* illustrée par la Figure 5.10.

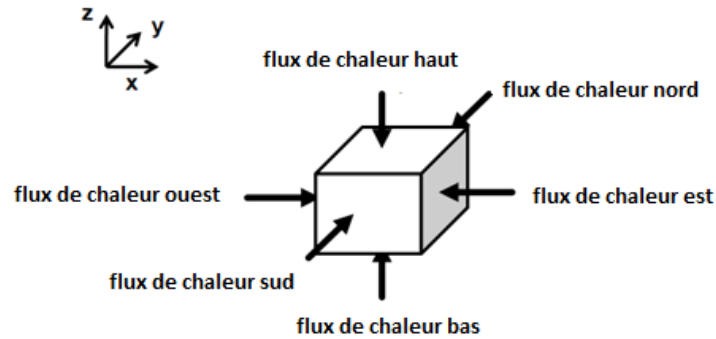


Figure 5.10: Flux de chaleur échangés dans une cellule [36]

Soit  $l$  la longueur,  $w$  la largeur et  $h_i$  la hauteur de la cellule  $Cell$ ,  $V$  son volume. Soit  $k$  et  $c$  les constantes de conductance thermique et capacité thermique du matériau, dont la cellule  $Cell$ , est fabriquée. La cellule  $Cell$  peut être faite de solide ou de liquide, dans le cas d'une cellule liquide  $h$  et  $u$  sont respectivement le coefficient de transfert thermique et le débit du liquide.

La valeur de capacitance de la cellule  $Cell$  est égale à la valeur de la capacité du matériau multipliée par le volume de celle-ci :  $Capacitance = c * V$

Les valeurs de conductance de la cellule  $Cell$  sont calculées à partir des échanges sur chacune de ses faces et le type du matériau dont elle est conçue. Pour simplifier, nous allons représenter la conductance de la cellule dans la direction Est, pour retrouver les valeurs des conductances dans les autres directions, on suit le même raisonnement.

La Figure 5.11 résume le calcul des valeurs de capacitance pour une cellule dans tous les cas de figure.

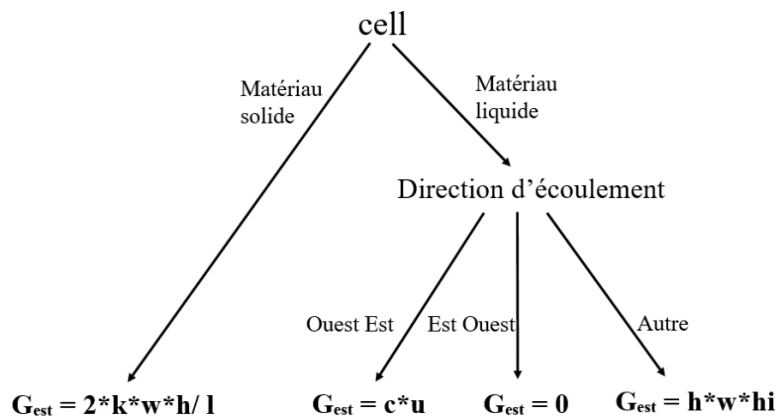


Figure 5.11: Calcul des capacitances pour une cellule

## 5.11 La résolution

Pour le cas d'un régime permanent, la résolution est dépendante de la composition du circuit. En absence de composants fluidiques, l'algorithme du gradient conjugué préconditionné est utilisé. Pour les circuits refroidis avec écoulement liquide, la résolution du système matriciel est assurée par la méthode GMRES.

Dans le cas d'un régime transitoire, l'algorithme de Thomas combiné à un schéma itératif de correction de l'erreur de décomposition est employé pour résoudre le système d'équations.

## 5.12 Le support de nouvelles options et fonctionnalités

Les circuits évolués intégrant des systèmes de refroidissement à base de liquide présentent des défis importants pour la simulation. Ainsi, la simulation doit permettre de :

- Explorer les valeurs de débit du fluide de refroidissement utilisé.
- Explorer la valeur du coefficient de transfert thermique afin de trouver ceux qui offrent un meilleur refroidissement.
- Considérer le débit du fluide variable selon la température d'un ensemble de cellules du circuit.

Pour assurer la généralisation de l'outil à ce type de circuit et permettre son évolution, ICTherm a été enrichi de telle sorte que :

- Certains paramètres sont ajoutés à la partie frontale de l'outil.
- La phase de discrétisation et de résolution des équations sera relancée jusqu'à convergence : la différence entre la température de deux itérations successives doit atteindre un certain epsilon.
- Un module a été ajouté pour le calcul du paramètre débit variable selon la température des cellules voisines et un algorithme de calcul.

## 5.13 Le module de sortie

L'étape de résolution décrite à la section 5.11 donne des valeurs de température pour toutes les cellules de maillage. Ces valeurs et les dimensions de chaque cellule correspondante sont passées à un module de sortie pour générer une carte thermique graphique.

La carte thermique facilite la visualisation et l'interprétation des résultats par les concepteurs.

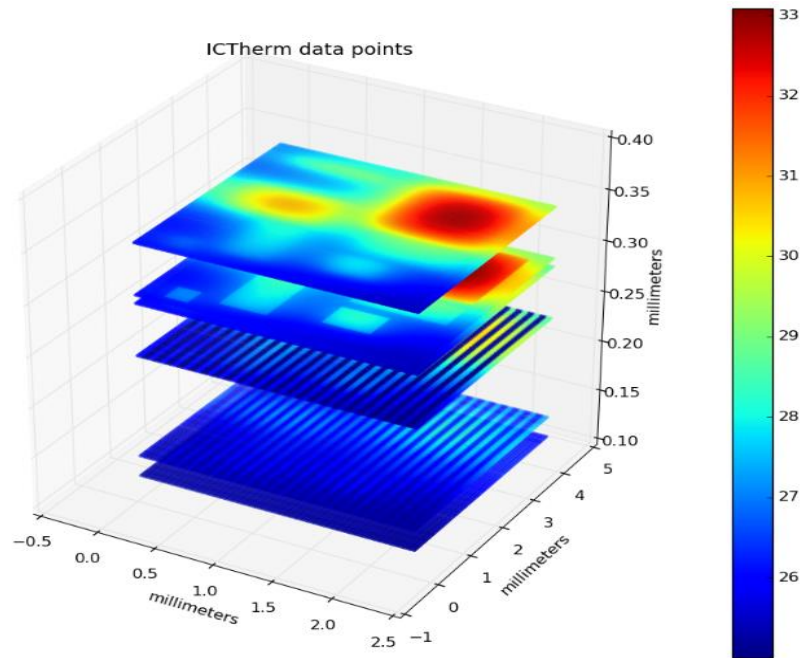


Figure 5.12: Carte thermique d'un circuit intégré tridimensionnel obtenu par sa simulation à travers l'outil ICTherm

Un exemple de carte thermique résultante d'une simulation d'un circuit par ICTherm est donné dans la Figure 5.12.

Le module de sortie graphique nous offre la possibilité de visualiser la température de chaque point dans le circuit en cliquant dessus, d'agrandir l'image ou la réduire, de la faire pivoter. Une carte thermique par couche est aussi possible pour aider le concepteur à mieux visualiser les résultats. Un fichier avec la température moyenne de chaque bloc décrit dans la modélisation, ainsi qu'un fichier donnant la température et les coordonnées de chaque cellule du maillage sont générés.



## CHAPITRE 6 RÉSULTATS EXPÉRIMENTAUX

Cette partie ne vise pas à valider les méthodologies utilisées par l'outil ICTherm et le choix des solveurs et leur efficacité, précédemment vérifiés par les travaux d'Alain Fourmigue [4]. Le but de cette partie est donc de montrer l'aspect générique de l'outil, l'aspect évolutif et sa facilité d'utilisation et mise à jour.

Les travaux seront validés par trois circuits :

- Un circuit représentant le processeur AtomD2700 [7].
- Un circuit microfluidique basé sur le processeur SPARC T1 [8].
- Un circuit intégrant des composants optiques.

### 6.1 Choix des circuits

Les circuits Atom D2700 et les circuits SPARC T1 sont des circuits fabriqués et exploités. Ils ont été choisis pour valider la précision de l'outil ICTherm et sa capacité à modéliser les types de circuits existants.

L'Atom D2700 est un circuit purement solide refroidi par un diffuseur de chaleur. Les flux de chaleur sont de type conductif et le système d'équations à résoudre est symétrique. Il sera simulé pour le régime permanent et la précision des résultats sera comparée à celle de l'outil commercial Comsol.

Le SPARC T1 est un circuit multicouche refroidi par écoulement de liquide, les transferts de chaleur se font par conduction et convection. Il est inspiré d'un modèle utilisé dans les travaux de Fourmigue et al. [36].

Le dernier circuit est un circuit optique qui est en cours de conception. La difficulté de ce circuit réside dans les dimensions très variées de ses composants. Ce circuit du futur est choisi pour valider l'aspect évolutif de l'outil ICTherm et sa capacité à modéliser les nouveaux types de circuits intégrés tridimensionnels.

## 6.2 Intel Atom D2700

Le premier circuit modélisé par ICTherm est un circuit purement solide : l'Intel Atom D2700. L'architecture du circuit et sa modélisation seront détaillées dans les sections qui suivent.

### 6.2.1 Description du circuit

L'Atom D2700 [7] est un processeur composé de deux cœurs, il est contenu dans une carte mère Intel BOXD2700DC, une cellule de Peltier est utilisée pour refroidir le circuit. La Figure 6.1 illustre le plan de l'Atom D2700.

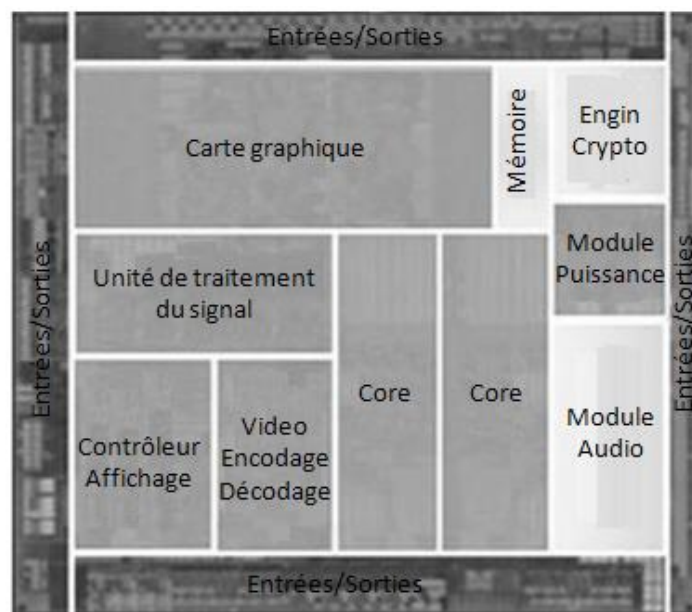


Figure 6.1: Plan de masse de l'Atom D2700 [7]

### 6.2.2 Modélisation

Pour effectuer la simulation de ce circuit, une modélisation des 10 composants le constituant a été faite par ICTherm et l'outil Comsol.

ICTherm se base sur un maillage orthogonal structuré conforme tandis que Comsol sur un maillage tétraédrique. Les détails sur les couches, les matériaux dont elles sont faits ainsi que leur épaisseur sont présentés dans le Tableau 6.1 et le modèle thermique utilisé pour les deux simulateurs est illustré par la Figure 6.2.

Tableau 6.1: Éléments du modèle thermique de l'Atom D2700

<i>Élément</i>	<i>Matériau</i>	<i>Épaisseur (<math>\mu\text{m}</math>)</i>
<i>Puce</i>	<i>Silicium</i>	<i>700</i>
<i>Couche active</i>	<i>Mélange métal-diélectrique</i>	<i>10</i>
<i>Billes C4</i>	<i>Alliage Étain-Plomb</i>	<i>100</i>
<i>Substrat</i>	<i>FR4</i>	<i>1000</i>
<i>Billes BGA</i>	<i>Alliage Étain-Plomb</i>	<i>300</i>
<i>Carte mère</i>	<i>FR4</i>	<i>1500</i>
<i>Pâte thermique</i>	<i>TIM</i>	<i>2000</i>
<i>Diffuseur de chaleur</i>	<i>Cuivre</i>	<i>4000</i>

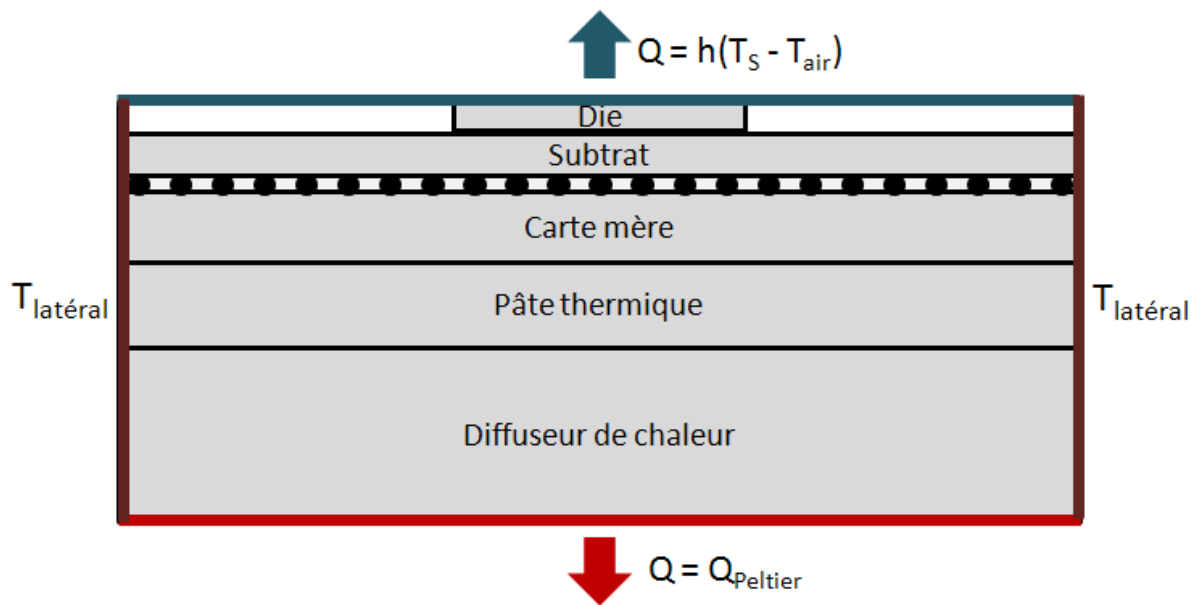


Figure 6.2: Modèle thermique proposé [37]

### 6.2.3 Simulation

Une fois le modèle réalisé, trois simulations du régime permanent ont été faites par ICTherm et Comsol. Dans un premier temps le processeur est simulé à 50% de sa capacité de calcul puis à 90% et enfin à 100%. Les résultats obtenus sont illustrés dans la Figure 6.3 et démontrent une précision

de l'outil ICTherm très similaire à l'outil commercial Comsol. La différence de température pour chaque scénario ne dépasse pas le  $0.13^{\circ}\text{C}$ .

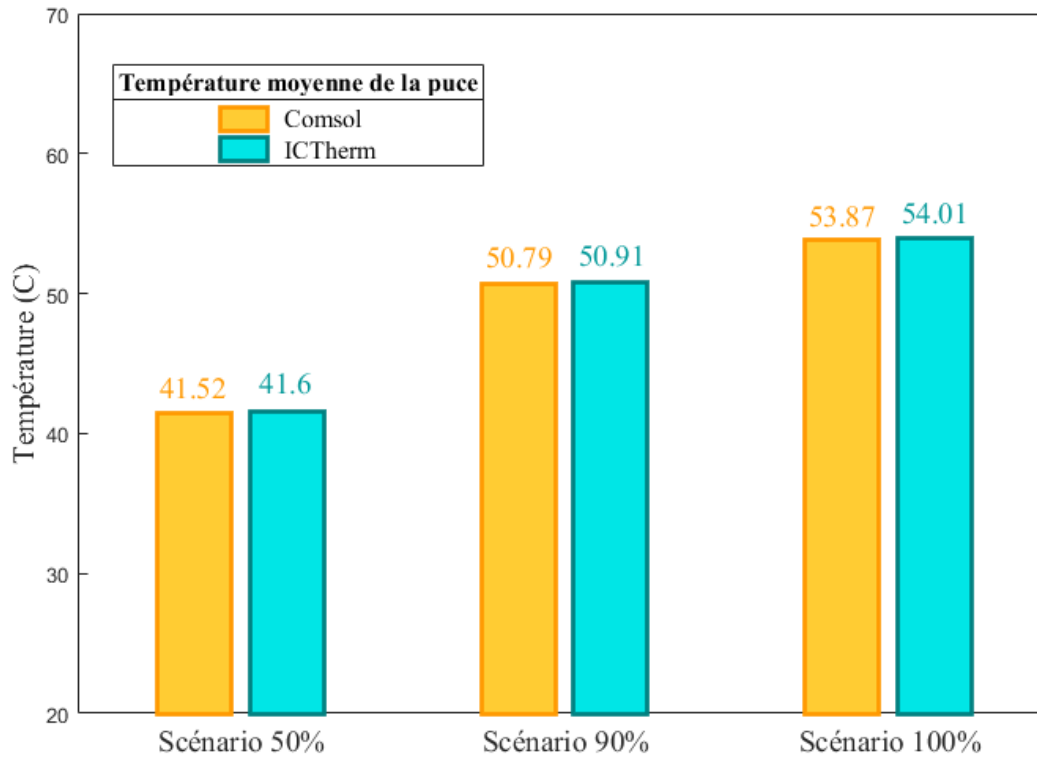


Figure 6.3: Comparaison des températures moyennes obtenues par simulation du régime permanent de l'Atom D2700 par les simulateurs Comsol et ICTherm pour une activité du processeur à 50%, 90 % et 100%

### 6.3 Circuit intégré tridimensionnel refroidi par écoulement liquide

Le circuit intégré étudié est un circuit tridimensionnel refroidi par écoulement liquide inspiré d'un modèle utilisé dans les travaux de Fourmigue et al. [36], basé sur le processeur SPARC T1.

Le liquide utilisé pour le refroidissement dans le cas de ce circuit est de l'eau.

Pour modéliser ce circuit, Fourmigue a utilisé un code en dur. Le code a été écrit pour le circuit en question, tous les paramètres et les éléments sont des valeurs constantes. Le code est assez rigide, pas adaptable et très contraignant. Les résultats obtenus seront comparés à ceux générés par ICTherm générique pour ce circuit intégré tridimensionnel.

### 6.3.1 Description du circuit

L'architecture du circuit est illustrée dans la Figure 6.4. Il est composé de quatre puces superposées, les puces 2 et 4 sont des puces contenant un cœur processeur, les puces 1 et 3 contiennent une mémoire L2 cache.

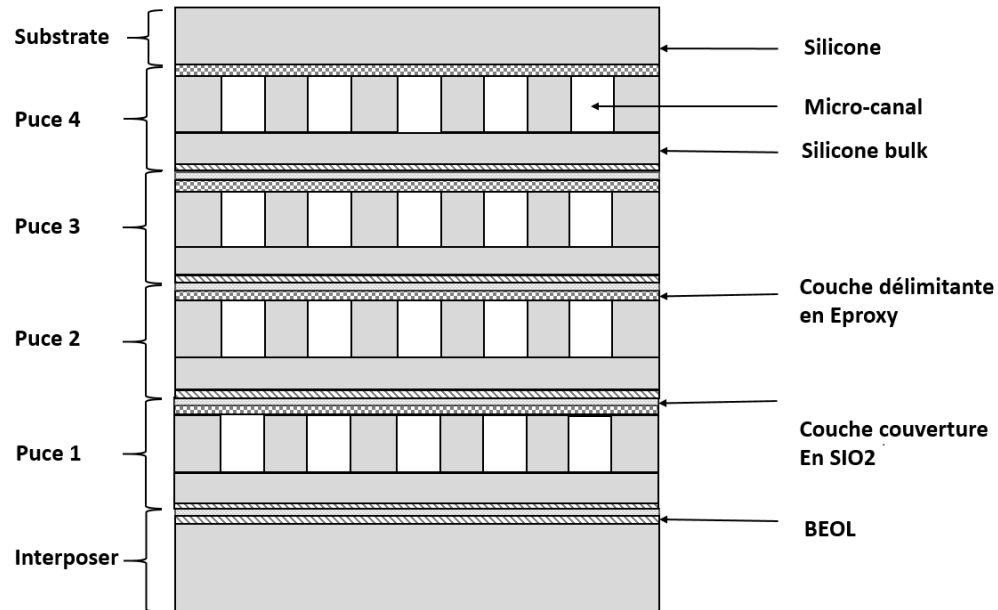


Figure 6.4: Circuit multicouche du premier test

La Figure 6.5 décrit l'architecture des cœurs des couches 2 et 4, qui disposent de 7 sources de chaleur qui dégagent en tout 3.075 W de puissance et les caches des couches 1 et 3 avec 2 sources de chaleur qui dégagent 0.792 W de puissance.

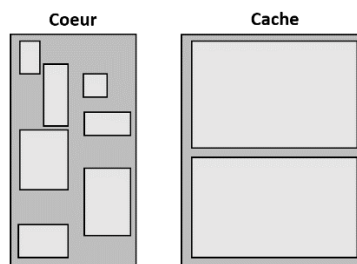


Figure 6.5: Disposition des sources de chaleur [36]

### 6.3.2 Modélisation

Les éléments du modèle thermique utilisés par ICTherm sont détaillés dans le Tableau 6.2.

Tableau 6.2: Éléments du modèle thermique

<b><i>Architecture</i></b>	
<i>Longueur de la pile (<math>\mu\text{m}</math>)</i>	<i>2350</i>
<i>Largeur de la pile (<math>\mu\text{m}</math>)</i>	<i>4200</i>
<i>Épaisseur de la pile (<math>\mu\text{m}</math>)</i>	<i>1096</i>
<i>Nombre de puces</i>	<i>4</i>
<i>Nombre de couches</i>	<i>23</i>
<b><i>Puce</i></b>	
<i>Couche silicone bulk</i>	<i>8</i>
<i>Couche métal</i>	<i>51</i>
<i>Couche couverture</i>	<i>20</i>
<i>Couche délimitante</i>	<i>10</i>
<b><i>Microcanaux</i></b>	
<i>Épaisseur</i>	<i>100</i>
<i>Longueur</i>	<i>50</i>
<i>Nombre par couche</i>	<i>23</i>

En tout, le circuit comporte 133 composants entre couches solides, sources de chaleur et canaux microfluidiques. Ces composants du circuit, les matériaux les constituant ainsi que les paramètres du circuit ont été facilement modélisés par les entités blocs, matériaux et simulation du module frontal d'ICTherm. Pour être fidèle aux conditions fixées dans le code en dur utilisé comme référence, le maillage est choisi régulier avec une taille de cellule de  $50\mu\text{m} \times 100\mu\text{m} \times 8\mu\text{m}$ . Des tests ont été faits avec un maillage irrégulier plus fin et donnent une meilleure précision.

### 6.3.3 Simulation

La simulation du régime transitoire est faite sur 0.1s avec 1000 pas de simulation. La température de l'air ambiant et du liquide est de 300K.

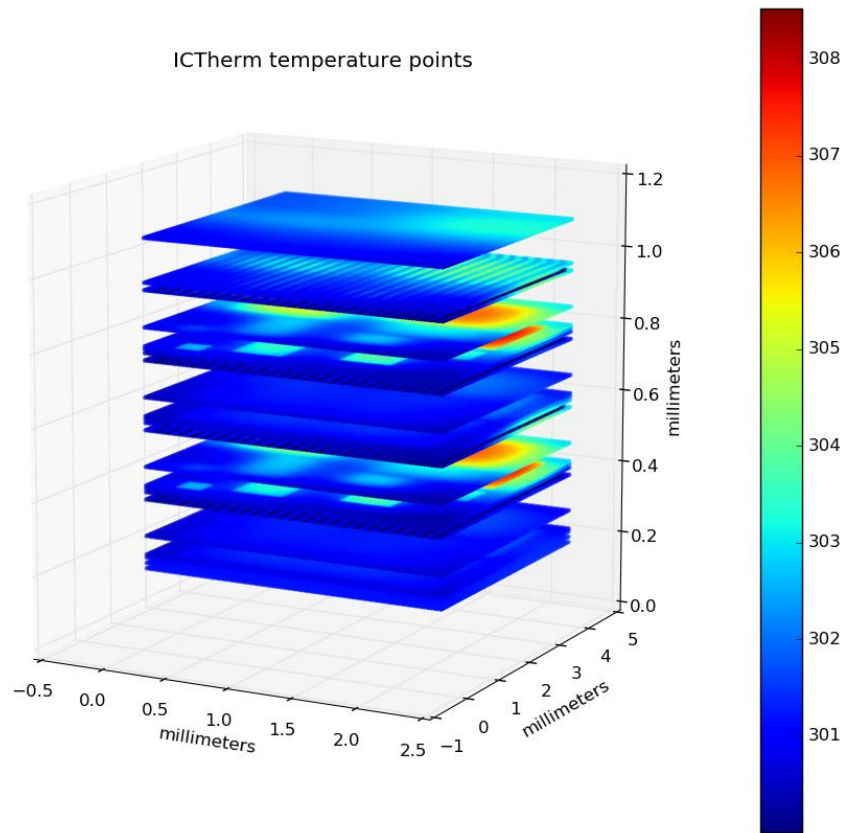


Figure 6.6: Carte thermique du circuit après simulation du régime transitoire

La Figure 6.6 montre la carte thermique réalisée avec l'outil ICTherm après la simulation, on y distingue les couches constituant le circuit ainsi que la variation de température en tout point. Une comparaison des températures par rapport aux résultats obtenus dans les travaux précédents a été réalisée, elle est illustrée dans la Figure 6.7. Une différence moyenne inférieure à 0.15 Kelvin est enregistrée entre les deux simulations. Cette différence s'explique par les arrondissements des calculs, l'implémentation des algorithmes, mais surtout par la considération dans ICTherm de puissance volumique à la place des puissances surfaciques précédemment considérées.

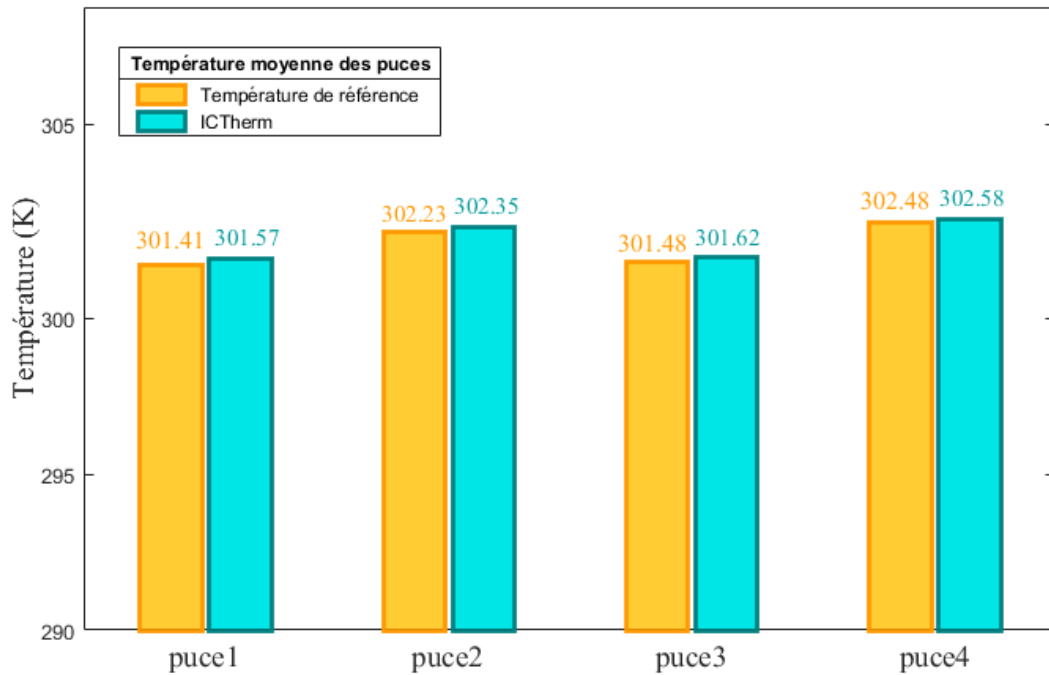


Figure 6.7: Comparaison des températures moyennes des quatre puces du circuit lors de la simulation par ICTherm et dans l'implémentation en dur

## 6.4 Circuit optique

Pour travailler sur l'aspect évolutif d'ICTherm et sa capacité à modéliser et simuler les circuits futurs nous avons collaboré avec Sébastien Le Beux et son équipe de l'université de Lyon pour travailler sur leur circuit optique en cours de conception.

### 6.4.1 Description du circuit

Comme décrit dans la Figure 6.8 et la Figure 6.9, le circuit contient un ensemble de couches électriques et des couches représentant un réseau sur puce (Network on Chip) optique constitué par des matrices de lasers.



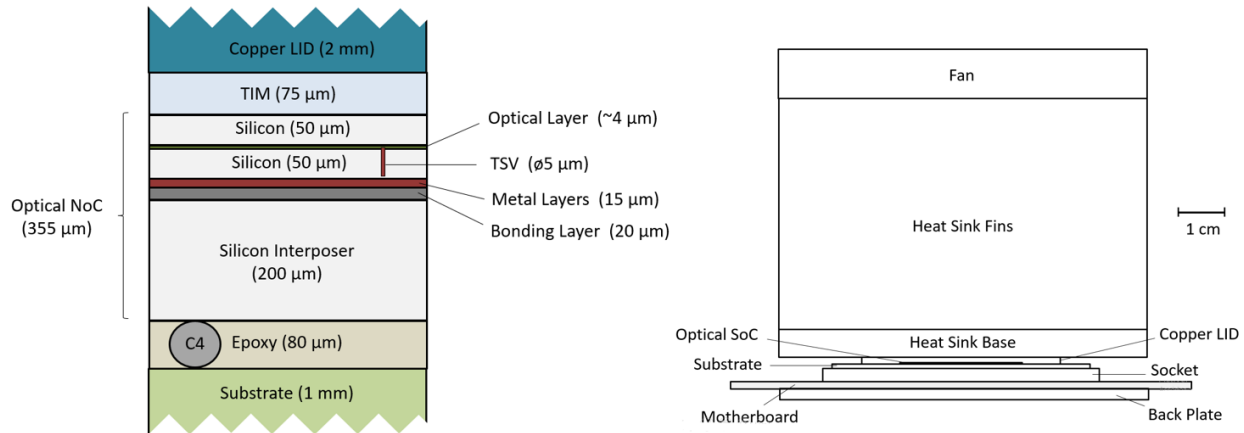


Figure 6.8: Architecture du circuit optique [34]

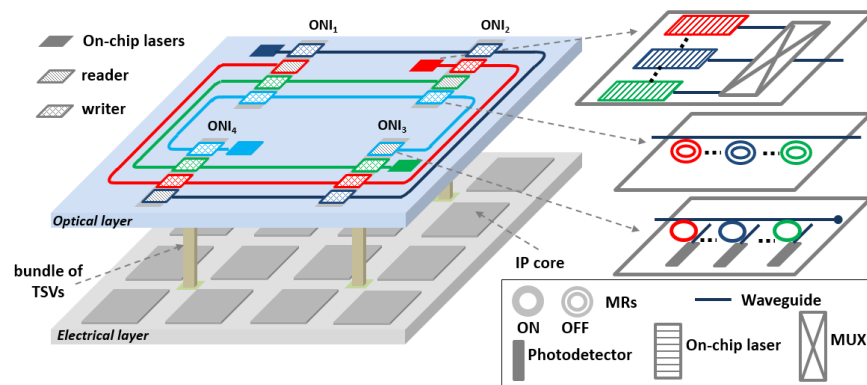


Figure 6.9: Architecture de la couche contenant les lasers [34]

## 6.4.2 Modélisation

Chaque bloc nommé ONI (venant de l'anglais Optical Network Interface) contient une matrice de lasers de dimensions  $30\mu\text{m} \times 15\mu\text{m} \times 4\mu\text{m}$ . Tous les composants du circuit (plus de 150 composants électriques et optiques) ont été modélisés par des blocs (voir définition dans la section 5.8) y compris les lasers qui sont considérés comme des sources de chaleur. Pour plus de précision, un maillage non régulier de plus de 824000 cellules a été utilisé, le maillage est plus fin autour des lasers.

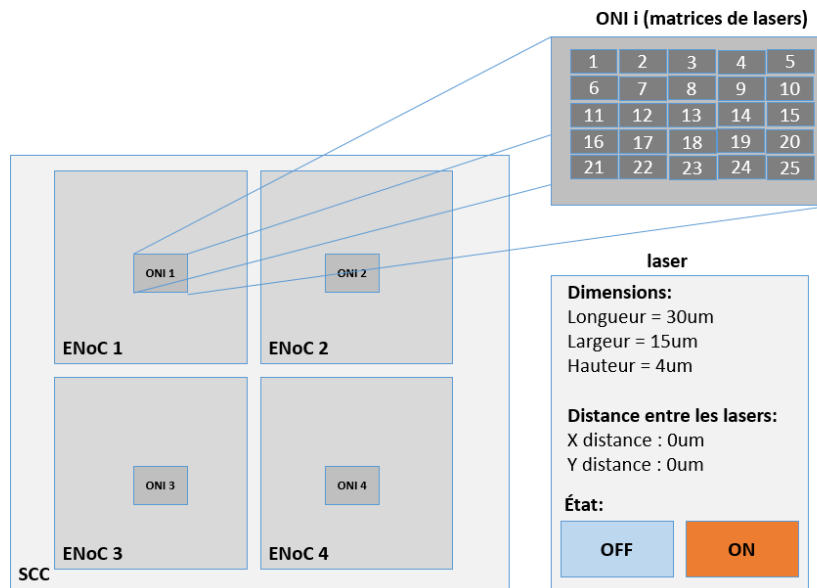


Figure 6.10: Architecture de la couche contenant les lasers

La Figure 6.10 décrit la topologie de la couche contenant les matrices de lasers.

### 6.4.3 Simulation

Pour voir l'effet de l'activation ou non des lasers sur le réchauffement du circuit et explorer les valeurs de puissances, six simulations en régime permanent ont été effectuées.

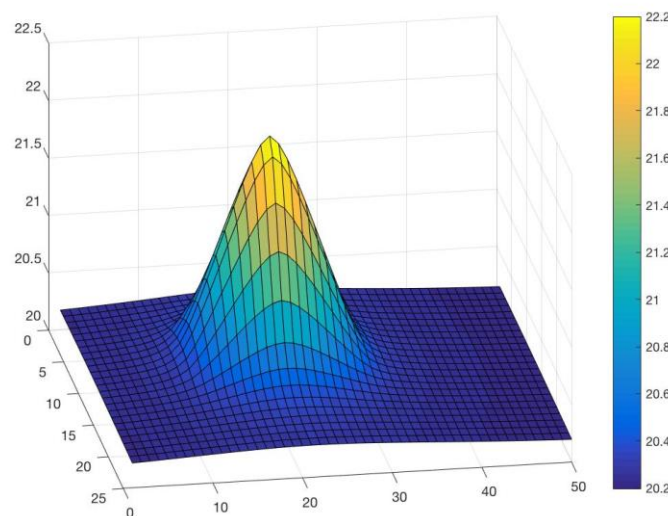


Figure 6.11: Distribution de la température dans la matrice des lasers lors de l'activation du laser numéro 13 avec une puissance de 10mW

Dans la matrice 5x5 des lasers, le laser numéro 13 a été allumé et des simulations avec des valeurs de puissances de 100μW, 500μW, 1mW, 2mW, 5mW et 10mW ont été réalisées<sup>2</sup>, les résultats sont donnés dans le Tableau 6.3.

Tableau 6.3: Valeurs de température minimale et maximale de la matrice des lasers à différentes valeurs de puissances.

<i><b>Puissance du laser allumé en mW</b></i>	<i><b>Température minimale de la matrice des lasers en °C</b></i>	<i><b>Température maximale de la matrice des lasers en °C</b></i>
<i>0.1</i>	<i>20.002</i>	<i>20.022</i>
<i>0.5</i>	<i>20.01</i>	<i>20.11</i>
<i>1</i>	<i>20.02</i>	<i>20.22</i>
<i>2</i>	<i>20.05</i>	<i>20.4</i>
<i>5</i>	<i>20.1</i>	<i>21.1</i>
<i>10</i>	<i>20.2</i>	<i>22.2</i>

## 6.5 Synthèse des résultats

Le travail de validation réalisé prouve la capacité de l’outil ICTherm à modéliser et simuler les circuits solides, microfluidiques ainsi que les circuits futurs tels que les circuits optiques. La précision des valeurs de température calculées par le simulateur est similaire à celle obtenue par l’outil Comsol et les résultats précédemment obtenus lors de la validation des méthodologies par Alain Fourmigue.

---

<sup>2</sup> Les simulations ont été réalisées en collaboration avec Yuqi Wang, étudiante à l’École Centrale de Lyon

## CHAPITRE 7 CONCLUSION

### 7.1 Synthèse

Les circuits intégrés tridimensionnels apportent divers avantages tels qu'une taille compacte, une grande bande passante et une latence réduite ; mais du à leur architecture en couches ils connaissent des problèmes de dissipation de chaleur qui engendre un réchauffement pouvant causer une dégradation de leurs fonctionnalités et une réduction de leur durée de vie. Les méthodes classiques de refroidissement peinent à garder ces circuits à une température acceptable, le recours à un refroidissement par écoulement liquide est alors essentiel ; mais cette technique ajoute de la complexité à la tâche de conception des circuits tridimensionnels. La simulation thermique offre la possibilité aux concepteurs de revoir leur conception et effectuer plusieurs itérations avant la phase de fabrication. La simulation des circuits intégrés tridimensionnels refroidis par écoulement liquide présente beaucoup de défis comme la complexité de l'architecture de ces circuits, l'hétérogénéité des matériaux les constituant, le besoin de couvrir le régime permanent ainsi que le régime transitoire, l'évolution continue des fonctionnalités et composants de ce type de circuits.

La littérature est riche de travaux qui se sont intéressés à la simulation thermique des circuits intégrés tridimensionnels ; plusieurs méthodes de modélisation et simulation existent, mais les méthodes numériques se distinguent par leur efficacité. Les méthodes thermiques numériques se basent sur une discrétisation spatiale et temporelle du système afin de ramener la résolution de l'équation différentielle de chaleur à la résolution d'un problème d'algèbre linéaire. Le système matriciel obtenu est sous la forme de  $Ax=b$  solvable par des méthodes directes ou itératives.

Les simulateurs thermiques existants basés sur des méthodes thermiques numériques se limitent à résoudre des systèmes de tailles moyennes, sont gourmands en mémoire et en temps d'exécution, payants ou ne permettent pas de suivre l'évolution rapide des circuits intégrés tridimensionnels et ne supportent pas de nouvelles fonctionnalités et options de ceux-ci.

ICTherm est un simulateur thermique qui se propose de résoudre les inconvénients précédemment cités. Notre travail s'inscrit dans la continuité des travaux de recherches sur la simulation thermique des circuits intégrés tridimensionnels effectués dans notre laboratoire de recherche HESL (Heterogeneous Embedded Systems Laboratory). Les méthodologies suivies dans ICTherm sont basées sur des modèles thermiques numériques. Une discrétisation spatiale par la méthode des

différences finies est réalisée sur un maillage orthogonal structuré conforme représentant le circuit. Pour la simulation du régime permanent, cette discrétisation est suffisante pour avoir une équation sous la forme  $GT=P$ , qui sera résolu par la méthode du gradient conjugué dans le cas d'un circuit fait de matériau solide et par l'algorithme GMRES dans le cas d'un circuit refroidi par écoulement de fluide. Pour le régime transitoire, une discrétisation temporelle à l'aide de la méthode d'Euler arrière est faite ; elle est suivie par décomposition des opérateurs sur les trois dimensions du circuit, le système matriciel tridiagonal obtenu est résolu par l'algorithme de Thomas puis un schéma itératif pour corriger l'erreur de décomposition est mis en place. Des implémentations précédentes de l'outil ICTherm ont été réalisées dans le but de valider les méthodologies utilisées. Ces implémentations présentent plusieurs limitations : dans certains cas il s'agit de code en dur pour modéliser un circuit en particulier, dans d'autres un code peu flexible pour simuler un type particulier de circuit pour le régime permanent.

Les travaux réalisés lors de cette maîtrise ont permis : d'une part, de généraliser les méthodes proposées à tout type de circuit intégré tridimensionnel existant (solide et microfluidique) pour les régimes permanent et transitoire et de proposer une modélisation efficace de ces circuits; d'autre part, d'adapter ces méthodologies pour permettre la simulation des futurs circuits intégrés en supportant de nouvelles fonctionnalités et en facilitant l'ajout et la mise à jour des modules et options présents. Pour réaliser ces objectifs :

- L'implémentation de l'outil ICTherm est faite en Python pour sa simplicité, sa proximité de Matlab, ses librairies très riches et son efficacité.
- Divers niveaux de maillage sont proposés pour offrir différents niveaux de précision.
- La modélisation des circuits est faite à travers un module qui gère la configuration de la simulation et de la topologie du circuit. Ce module offrant un format de données normalisé permet de décrire et donc de modéliser tout type de circuit.
- De nouvelles options et fonctionnalités sont offertes pour permettre la simulation des circuits futurs.
- Le code est basé sur une architecture modulaire permettant une extensibilité et une mise à jour simple et rapide.

- Un module d'entrée graphique, ergonomique et facile à utiliser ainsi qu'un module de sortie avec une représentation en carte thermique des résultats enrichissent l'outil ICTherm pour faciliter la saisie des données et l'interprétation des sorties.

L'outil ICTherm est validé par rapport à l'outil Comsol et des circuits préalablement testés dans de précédents travaux dans notre laboratoire. Trois circuits différents ont été utilisés pour vérifier l'exactitude des résultats obtenus par ICTherm ainsi que sa capacité à modéliser de nouveaux types de circuits tels que les circuits optiques. Les deux premiers circuits simulés représentent des processeurs réels existants, les résultats démontrent l'aptitude de l'outil ICTherm à simuler des circuits solides et microfluidiques pour le régime permanent et transitoire avec une précision similaire aux simulateurs du marché avec une différence qui ne dépasse pas  $0.5^{\circ}\text{C}$ . Le dernier circuit optique simulé montre la capacité de l'outil ICTherm à modéliser les circuits du futur.

## 7.2 Limitations

Une première limitation des travaux effectués est liée à la forme du maillage qui doit être orthogonal structuré conforme pour pouvoir appliquer la méthode de discrétisation spatiale choisie : méthode des différences finies.

La deuxième limitation est la nécessité de garder la structure physique du circuit et le pas de simulation invariable durant la simulation du régime transitoire. Il ne s'agit toutefois pas de contrainte majeure puisque ces paramètres ne sont pas censés changer au cours de simulation.

## 7.3 Améliorations et travaux futurs

Plusieurs travaux futurs possibles sont à suggérer pour améliorer divers aspects de l'outil ICTherm.

Une première amélioration sera de travailler sur le maillage adéquat pour augmenter la précision des calculs tout en gardant un temps d'exécution raisonnable, Abdelkader et al [38] propose une technique adaptative de choix de maillage, basée sur une phase pré caractérisation et un algorithme non itératif. Il serait intéressant d'étudier l'applicabilité et l'efficacité de leur méthodologie.

Une seconde amélioration concernera la précision des modèles définis en révisant les simplifications faites à l'équation de transfert par advection afin de mieux représenter les variations de vitesses des fluides considérés.

Une troisième amélioration possible touchera l'aspect graphique de l'outil, il faudra pousser encore plus loin les travaux apportés sur les modules entrée et sortie pour accentuer la facilité d'utilisation et l'ergonomie.

Une autre amélioration permettra la facilité d'ajout et de mise à jour des fonctionnalités et caractéristiques des circuits intégrés de nouvelles générations en offrant à l'utilisateur la possibilité de modéliser ses propres composants.

Une amélioration portera sur la parallélisation du code pour augmenter les performances de l'outil.

Une dernière amélioration pourra toucher le fichier intermédiaire utilisé pour modéliser le système, il pourra être plus généralisé et remplacé par un format plus standard tel qu'IP-XACT.

## RÉFÉRENCES

- [1] A. Vassighi et M. Sachdev, "Power, Junction Temperature, and Reliability," dans *Thermal and Power Management of Integrated Circuits*, 1<sup>e</sup> éd., S. US, Édit., 2006, p. 14-49.
- [2] R. S. Patti, "Three-dimensional integrated circuits and the future of system-on-chip designs," *Proc. IEEE*, vol. 94, n<sup>o</sup>. 6, p. 1214-1224, 2006. [En ligne]. Disponible: <https://doi.org/10.1109/JPROC.2006.873612>
- [3] V. F. Pavlidis et E. G. Friedman, "Thermal Management Techniques", dans *Three-dimensional integrated circuit design*: Morgan Kaufmann, 2009, p. 99-133. [En ligne]. Disponible: <https://doi.org/10.1016/B978-0-12-374343-5.00006-X>
- [4] A. Fourmigue, "Méthodes de calcul numérique pour la simulation thermique des circuits intégrés," École polytechnique de Montréal, Montréal, QC, 2014. [En ligne]. Disponible: [http://www.icherm.com/these\\_alain\\_fourmigue.pdf](http://www.icherm.com/these_alain_fourmigue.pdf)
- [5] S. G. Kandlikar, D. Kudithipudi et C. A. Rubio-Jimenez, "Cooling mechanisms in 3D ICs: Thermo-mechanical perspective," communication présentée à International Green Computing Conference (IGCC), Orlando, FL, United states, 2011. [En ligne]. Disponible: <http://dx.doi.org/10.1109/IGCC.2011.6008573>
- [6] S. Zimmermann *et al.*, "Aquasar: A hot water cooled data center with direct energy reuse," *ENERGY*, vol. 43, n<sup>o</sup>. 1, p. 237-245, 2012. [En ligne]. Disponible: <https://doi.org/10.1016/j.energy.2012.04.037>
- [7] Intel. (2014) Atom D2700. [En ligne]. Disponible: [https://ark.intel.com/products/59683/Intel-Atom-Processor-D2700-1M-Cache-2\\_13-GHz](https://ark.intel.com/products/59683/Intel-Atom-Processor-D2700-1M-Cache-2_13-GHz)
- [8] A. S. Leon *et al.*, "A power-efficient high-throughput 32-thread SPARC processor," *IEEE J. Solid-State Circuits*, vol. 42, n<sup>o</sup>. 1, p. 7-16, 2007. [En ligne]. Disponible: <https://doi.org/10.1109/JSSC.2006.885049>
- [9] N. H. E. Weste et D. F. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 3<sup>e</sup> éd. Boston, US: Pearson/Addison-Wesley, 2005.
- [10] J. H. Lau, "Evolution, challenge, and outlook of TSV, 3D IC integration and 3d silicon integration," communication présentée à International Symposium on Advanced Packaging Materials (APM), Xiamen, China, 2011, p. 462-488. [En ligne]. Disponible: <http://dx.doi.org/10.1109/ISAPM.2011.6105753>
- [11] J. Howard *et al.*, "A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling," *IEEE J. Solid-State Circuits*, vol. 46, n<sup>o</sup>. 1, p. 173-83, 01/ 2011. [En ligne]. Disponible: <http://dx.doi.org/10.1109/JSSC.2010.2079450>
- [12] J. P. Holman, *Heat Transfer*, 9<sup>e</sup> éd. Montreal, QC: McGraw-Hill, 2002.
- [13] T. L. Bergman *et al.*, *Fundamentals of Heat and Mass Transfer*, 7<sup>e</sup> éd. Hoboken, N.J: John Wiley & Sons, 2011.
- [14] Y.-K. Cheng *et al.*, "Thermal Simulation for VLSI Systems", dans *Electrothermal Analysis of VLSI Systems*, Boston, MA: Springer US, 2002, p. 61-93. [En ligne]. Disponible: [http://dx.doi.org/10.1007/0-306-47024-1\\_4](http://dx.doi.org/10.1007/0-306-47024-1_4)



- [15] V. Szekely *et al.*, "THERMAN: A thermal simulation tool for IC chips, microstructures and PW boards," *Microelectronics Reliability*, vol. 40, n°. 3, p. 517-524, 2000.
- [16] Y.-M. Lee *et al.*, "NUMANA: A hybrid numerical and analytical thermal simulator for 3-D ICs," communication présentée à 16th Design, Automation and Test in Europe Conference and Exhibition (DATE), Grenoble, France, 2013, p. 1379-1384.
- [17] S. Shidore. (2007) Compact thermal modeling in electronics design. [En ligne]. Disponible: <https://www.electronics-cooling.com/2007/05/compact-thermal-modeling-in-electronics-design/>
- [18] H. Rosten et C. Lasance, "DELPHI: The Development of Libraries of Physical Models of Electronic Components for an Integrated Design Environment," dans *Model Generation in Electronic Design*, vol. 1, S. US, Édit., 1995, p. 63-89.
- [19] W. Huang *et al.*, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, n°. 5, p. 501-513, 2006. [En ligne]. Disponible: <http://ieeexplore.ieee.org/document/1650228/>
- [20] W. H. Press *et al.*, "Numerical Recipes in C++: The Art of Scientific Computing," *Eur. J. Phys.*, vol. 24, n°. 3, p. 329-330, 2003. [En ligne]. Disponible: <http://iopscience.iop.org/article/10.1088/0143-0807/24/3/701/meta>
- [21] H. Mizunuma, Y.-C. Lu et C.-L. Yang, "Thermal modeling and analysis for 3-D ICs with integrated microchannel cooling," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, n°. 9, p. 1293-1306, 2011. [En ligne]. Disponible: <http://ieeexplore.ieee.org/document/5989994/>
- [22] B. Goplen et S. S. Sapatnekar, "Placement of thermal vias in 3-D ICs using various thermal objectives," communication présentée à Proc. Int. Conf. Computer-Aided. Design (ICCAD), 2006, p. 692-708. [En ligne]. Disponible: <http://dx.doi.org/10.1109/TCAD.2006.870069>
- [23] X.-X. Liu *et al.*, "Full-chip thermal analysis of 3D ICs with liquid cooling by GPU-accelerated GMRES method," communication présentée à 13th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, United states, 2012, p. 123-128. [En ligne]. Disponible: <http://dx.doi.org/10.1109/ISQED.2012.6187484>
- [24] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2<sup>e</sup> éd. Philadelphia, Pa: Society for Industrial and Applied Mathematics, 2003.
- [25] T.-Y. Wang et C. C.-P. Chen, "3-D thermal-ADI: A linear-time chip level transient thermal simulator," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, n°. 12, p. 1434-1445, 2002. [En ligne]. Disponible: <http://dx.doi.org/10.1109/TCAD.2002.804385>
- [26] A. Sridhar *et al.*, "3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling," communication présentée à IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, United states, 2010, p. 463-470. [En ligne]. Disponible: <http://dx.doi.org/10.1109/ICCAD.2010.5653749>
- [27] S. C. Chapra et R. P. Canale, *Numerical Methods for Engineers*, 5<sup>e</sup> éd. Montréal, QC: McGraw-Hill Higher Education, 2006.

- [28] A. Vincenzi *et al.*, "Accelerating thermal simulations of 3D ICs with liquid cooling using neural networks," communication présentée à Proc. Int. Conf. Computer-Aided. Design (ICCAD), San Jose, USA, 2012 p. 15-20. [En ligne]. Disponible: <http://dl.acm.org/citation.cfm?id=2206787>
- [29] H. Qian, C.-H. Chang et H. Yu, "An efficient channel clustering and flow rate allocation algorithm for non-uniform microfluidic cooling of 3D integrated circuits," *Integration, the VLSI Journal*, vol. 46, n°. 1, p. 57-68, 2013. [En ligne]. Disponible: <http://dx.doi.org/10.1016/j.vlsi.2011.12.005>
- [30] P. Li *et al.*, "IC thermal simulation and modeling via efficient multigrid-based approaches," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 25, n°. 9, p. 1763-1776, 2006. [En ligne]. Disponible: <http://dx.doi.org/10.1109/TCAD.2005.858276>
- [31] Comsol Inc. (2017) Comsol Multiphysics. [En ligne]. Disponible: <https://www.comsol.com/>
- [32] ANSYS Inc. (2017) [En ligne]. Disponible: <http://www.ansys.com/>
- [33] H. Azarkish *et al.*, "Self-adaptive microvalve array for energy efficient fluidic cooling in microelectronic systems," communication présentée à 16th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), Orlando, FL, USA, 2017, p. 522-529. [En ligne]. Disponible: <http://ieeexplore.ieee.org/abstract/document/7992531/>
- [34] H. Li *et al.*, "Towards maximum energy efficiency in nanophotonic interconnects with thermal-aware on-chip laser tuning," *IEEE Trans. Emerg. Topics Comput.*, p. 1-1, 2016. [En ligne]. Disponible: <http://ieeexplore.ieee.org/document/7465823/>
- [35] K. D. Lee, *Python Programming Fundamentals*. London, Angleterre: Springer, 2011, Disponible: <https://link.springer.com/book/10.1007%2F978-1-84996-537-8>
- [36] A. Fourmigue, G. Beltrame et G. Nicolescu, "Explicit transient thermal simulation of liquid-cooled 3D ICs," communication présentée à 16th Design, Automation and Test in Europe Conference and Exhibition (DATE), Grenoble, France, 2013, p. 1385-1390. [En ligne]. Disponible: <http://ieeexplore.ieee.org/document/6513729/>
- [37] S. Kannan *et al.*, "Physics-based low-cost test technique for high voltage LDMOS," vol. 29, n°. 6, p. 745-762, 2013. [En ligne]. Disponible: <http://dx.doi.org/10.1007/s10836-013-5417-5>
- [38] S. Abdelkader, A. ElRouby et M. Dessouky, "Power pre-characterized meshing algorithm for finite element thermal analysis of integrated circuits," communication présentée à 20th Design, Automation and Test in Europe (DATE), Swisstech, Lausanne, Switzerland, 2017, p. 1293-1296. [En ligne]. Disponible: <http://ieeexplore.ieee.org/document/7927192/>